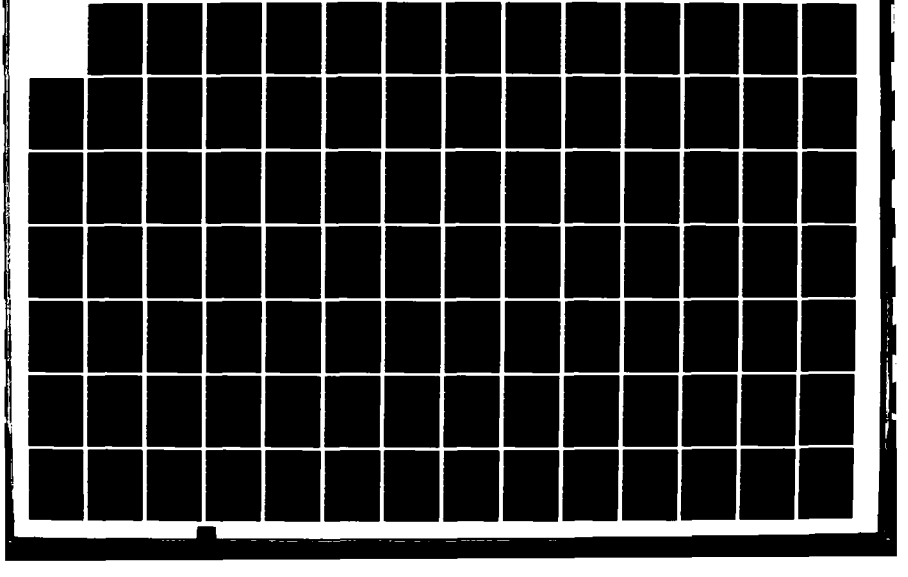
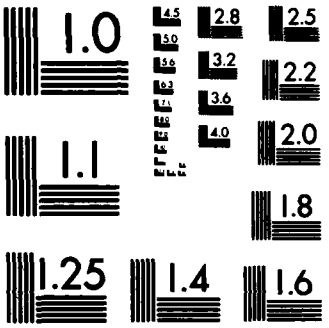
AD-A132 491 BANK-TO-TURN CONTROL(U) AIR FORCE INST OF TECH 1/2
WRIGHT-PATTERSON AFB OH D J CAUGHLIN 1983
AFIT/CI/NR-83-25T

UNCLASSIFIED F/G 16/4.1 NL





UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1 REPORT NUMBER: AFIT/CI/NR 83-25T	2 GOVT ACCESSION NO
3 RECIPIENT'S CATALOG NUMBER	
4 TITLE (and Subtitle): Bank-To-Turn Control	5 TYPE OF REPORT & PERIOD COVERED: THESIS/DISSERTATION
6 PERFORMING ORG REPORT NUMBER	
7 AUTHOR(s): Donald J. Caughlin, Jr.	8 CONTRACT OR GRANT NUMBER(s)
9 PERFORMING ORGANIZATION NAME AND ADDRESS: AFIT STUDENT AT: University of Florida	10 PROGRAM ELEMENT PROJECT TASK AREA & WORK UNIT NUMBERS
11 CONTROLLING OFFICE NAME AND ADDRESS: AFIT/NR, WPAFB OH 45433	12 REPORT DATE: 1983
	13 NUMBER OF PAGES: 100
14 MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15 SECURITY CLASS. (of this report): UNCLASS
	15a DECLASSIFICATION DOWNGRADING SCHEDULE

16 DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17 DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18 SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE: IAW AFR 190-17

1 SEP 1983

Approved for public release: IAW AFR 190-17.

LYNN E. WOLAVER

Dean for Research and Professional Development

Air Force Institute of Technology (ATC)

Wright-Patterson AFB OH 45433

19 KEY WORDS (Continue on reverse side if necessary and identify by block number)

20 ABSTRACT (Continue on reverse side if necessary and identify by block number)

ATTACHED

SEP 1 1983

E

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD A132491

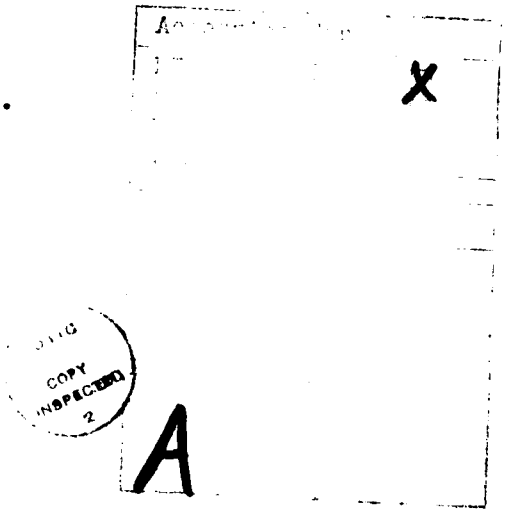
DTIC FILE COPY

83 09 14 080

BANK-TO-TURN CONTROL

BY

DONALD J. CAUGHLIN, JR.



A THESIS PRESENTED TO THE GRADUATE COUNCIL
OF THE UNIVERSITY OF FLORIDA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

1983

ACKNOWLEDGMENTS

The author wishes to express his gratitude to his committee chairman, Dr. T.E. Bullock, for his instruction, helpful suggestions, and encouragement. Appreciation is also expressed to the other committee members, Dr. C.V. Shaffer, Dr. J.R. Smith, and G. Basile. I am also deeply grateful to my wife, Barbara, whose understanding and support made this effort possible.

TABLE OF CONTENTS

LIST OF FIGURES

KEY TO SYMBOLS

Amax	Missile acceleration during the boost phase.
Amin	Missile deceleration during the coast phase.
Aoff	Spherical angle between the missile velocity vector and the line of sight to the target.
A_T	Target acceleration.
a	Maximum value of the control.
F	Matrix describing the dynamic interaction between the state variables.
F_x, F_y, F_z	Components of applied forces on respective body axes.
G_x, G_y, G_z	Components of applied moments on respective body axes.
g	Acceleration due to gravity.
H	Hamiltonian function.
I_x, I_y, I_z	Moment of inertia with respect to the given axis.
J	Cost function.
L	Matrix describing the dynamic interaction between the state and control variables.
m	Missile mass.
N_x, N_y, N_z	Components of applied acceleration on respective body axes.
p,q,r	Angular rates about the x, y, and z body axes, respectively.
p	Adjoint variables.

PITCH	The angle measured in the vertical plane between the x body axis and the inertial XY plane.
R	Slant range from missile to target.
ROLL	(Bank) Angle measured in the yz plane of the body system between the y body axis and the YX inertial plane.
T_b	Time during which the missile is accelerating (burn time).
Tgo	Time-to-go (Estimated).
t	Time.
t_f	Final time.
t_s	Switching time between saturated and unsaturated control.
t_0	Initial time.
u,v,w	Linear velocities with respect to the x, y, and z body axes, respectively.
u	Control variable.
V	Missile total velocity.
Vc	Closing velocity.
V_T	Target total velocity.
V_x,V_y,V_z	Inertial relative velocities with respect to the X, Y, and Z inertial axes, respectively.
x	State vector.
x,y,z	Body stabilized axes.
X,Y,Z	Inertial axes.
y	Terminal boundary condition.
YAW	The angle between the projection of the x body axis onto the inertial XY plane and the X inertial axis.
ϕ	Missile ROLL Euler angle.
θ	Missile PITCH Euler angle.

θ_a	Initial azimuth angle between the target velocity and missile LOS.
θ_g	Seeker elevation angle.
ψ	Missile YAW Euler angle.
ψ_a	Initial elevation angle between the target velocity and missile LOS.
ψ_g	Seeker azimuth angle.
ω_y, δ_z	Components of missile angular rates with respect to the respective seeker gimbal axes.
λ_T	Reciprocal of the target maneuver time constant.
γ	Weighting factor.

Abstract of Thesis Presented to the Graduate Council
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

BANK-TO-TURN CONTROL

By

Donald J. Caughlin, Jr.

December 1983

Chairman: Thomas E. Bullock
Major Department: Electrical Engineering

Linear optimal control laws used for bank-to-turn missile guidance were investigated. The performance of these control laws was erratic because their development neglected attitude dependent constraints on the control variables. The effects of an inequality constraint on the control variable in a fixed final time, zero error controller were analyzed. This analysis led to the conclusion that the accepted technique of applying the constraint to the unconstrained optimal control solution is valid only if the control is saturated during the initial portion of the trajectory. Constraints during the terminal phase must be anticipated by adjusting intermediate boundary conditions. The nonzero set point optimal control law used for the analysis of inequality constraints was incorporated into a nonlinear Bank-to-turn "attitude" controller. This controller, which was evaluated on a six degree of freedom simulation, successfully eliminated the deficiencies of the linear controllers.

T. E. Bullock

Chairman

CHAPTER I
INTRODUCTION

The classical approach for air-to-air missile control was based on velocity pursuit guidance, which is implemented by requiring the missile velocity vector to point at the target [1]. While adequate for slow moving targets, the concept fails in the current air-to-air scenario because, as the distance to the target decreases, the turning rate of the missile increases until unattainable accelerations may be required.

Current guidance laws are based on proportional navigation which causes the missile to fly a straight line trajectory toward the target [2,3,4]. In proportional navigation (pro-nav), the line-of-sight (LOS) rate is regulated to zero. Consequently, the implementation is simple because the outputs of a gimballed seeker provide direct missile acceleration commands. Also, this guidance law has better performance if the target and missile have constant speeds. Bryson and Ho [5] have shown that proportional guidance is optimal in the linear quadratic gaussian (LQG) sense if:

1. the target has constant velocity,
2. the missile has unlimited and instantaneous response,
3. the LOS angles are small,
4. the missile velocity along the LOS is constant.

However, these assumptions, inherent in the formulation, lead to rather serious limitations or inadequate performance in the current air-to-air arena [6]. Therefore, improved missile guidance laws are required for

more demanding engagements, more maneuverable targets, or improved accuracy.

While a number of guidance laws have been devised to improve the shortcomings of pro-nav [3,4,6,7,8,9], a number of advanced guidance techniques have been investigated to increase the launch envelope, provide better maneuver capability, and reduce the miss distance experienced with the classical approaches. These techniques use optimal control methods based on [1]:

1. Linear Quadratic Theory,
2. Linear Quadratic Gaussian Theory,
3. Singular Perturbation Theory,
4. Reachable Set Theory,
5. Differential Game Theory,
6. Adaptive Control,
7. Dual Control,
8. Spline Polynomial Approximations.

In general, these studies have either neglected the acceleration constraints, uncoupled the equations of motion, or reduced the problem to two dimensions. Nevertheless, a number of these optimal guidance laws have been applied to the study of a specific "generic" short range, highly maneuverable, bank-to-turn, air-to-air missile [10,11].

For the models under consideration, the optimal control laws were based on the finite dimensional linear system:

$$\dot{x} = Fx + Lu$$

where the state variables, and system matrices are defined to incorporate assumed system dynamics. The control laws were solved in inertial coordinates with a quadratic cost that included a penalty on miss distance and assumed a known fixed final time. Four linear and four nonlinear laws were defined by Fiske [8]. Table 1 presents the description of the linear models.

TABLE 1.1 Optimal Linear Guidance Laws

DESIGNATION	DESCRIPTION
G1	Optimal Linear Guidance.
G2	Optimal Linear Guidance accounting for target acceleration.
G3	Optimal Linear Guidance accounting for target acceleration and first-order missile/autopilot dynamics.
G4	Optimal Linear Guidance accounting for target acceleration and second-order missile/autopilot dynamics.

The nonlinear models were determined from the same models as the linear guidance laws except that the results were passed through the saturation function.

Initial results with the advanced guidance laws indicated only marginal performance gains over the classical technique, pro-nav. In fact, proportional guidance performed as well as optimal guidance where the missile had large heading errors (in violation of an assumption for pro-nav optimality) [5,8]. Also, increasing the complexity of the guidance law or filter (15 vs 9 states) did not statistically improve the performance [8].

As a simplifying assumption, the optimal control derivations assumed a fixed final time. And, since the control law is sensitive to this parameter, an initial explanation for the failure of the optimal guidance laws was the accuracy of the Time-to-go (Tgo) estimate [11]. Consequently, an improved Time-to-go estimating technique was developed by Riggs [12] . This technique was simple, stable, and effective in increasing the missile's performance on the inner launch boundary. The technique also addressed one of the other less desirable assumptions in

the formulation of the problem, control over axial acceleration. In effect, Riggs' Tgo algorithm operates by equating the x commanded acceleration from the control law to an estimate of the average acceleration and solves the unknown parameter Tgo. With this new estimate of Tgo, the advanced guidance laws (evaluated with a passive seeker) significantly outperformed pro-nav in the short range environment [11].

This thesis addresses some remaining limitations of the current optimal control laws and provides an alternative mechanization of optimal control results.

CHAPTER II
OPTIMAL CONTROL LAWS

Modern control theory [5,13,14] provides techniques to optimize guidance laws for air-to-air missiles. With respect to the stochastic control problem, Shalom and Tse [15] outline the various types of stochastic control laws available. Shalom and Tse and Van De Water and Willems [16] provide justification for the use of the Certainty Equivalence Property (no dual effect of second order). Consequently, although stochastic models are discussed in [5] and a specific technique is introduced in [8], the general procedure in air-to-air guidance is the use of filtered estimates that provided conditional expectation and rely on a dynamic-programming-like definition of optimality with the Certainty Equivalence Property to find optimal control laws [15,16,17]. Therefore, the guidance laws that have been derived and implemented in [8,10,11] and that will be considered here are deterministic.

As mentioned previously, the studies that have provided optimal (both linear and nonlinear) guidance laws have neglected the acceleration constraints, uncoupled the equations of motion, or reduced the problem to two dimensions. The application of these guidance laws to the bank-to-turn, air-to-air missile has produced unexpected results [10,11]. Most of the current anomolies can be traced to the formulation of the control law and the properties of the bank-to-turn missile.

While this vehicle can attain high normal accelerations, it has some unusual limitations, and the guidance laws did not consider these restrictions. To begin with, the control laws were derived using an unconstrained, fixed final time controller with a quadratic penalty function on the terminal error [7,8]. Actually, the problems are a terminal controller with zero terminal error, inequality constraints on the admissible controls, and final time free.

Also, there is an aspect of this generic bank-to-turn missile that is in conflict with the unconstrained formulation of the advance guidance laws. In contrast to the "standard" air-to-air missile which is symmetric and can maneuver equally well along either axis, this missile has a planar body that provides a high coefficient of lift. (See Appendix D.) Consequently, it can maneuver well along one axis (100 "g") and very poorly along the other (5 "g"), and in order to utilize it's acceleration potential, the missile must bank to maneuver (turn). The current control laws function with relative positions and velocities in inertial space. Yet, the control constraints are significantly different for each body axis and are fixed with respect to the missile body axis. The result is unequal control constraints that are a function of the relative target-missile attitude and geometry.

A related shortcoming of the guidance laws is the absence of roll control. This absence can lead to additional problems. First, the seeker gimbals are limited (60 degrees), and this limitation is not considered. Also, since the guidance laws rely on the autopilot to roll the missile, it may be possible for an evading target to maneuver itself such that its maximum maneuver direction is orthogonal to the missile's primary axis and gain a temporary maneuver advantage. Since the warhead

effects are a function of roll angle, these effects will couple with the maneuver advantage to limit the effectiveness of the missile.

The first shortcoming, the zero error controller versus the quadratic penalty on the miss distance, was first addressed by Youngblood [9]. In [9], the following assumptions were incorporated into the model:

1. PITCH and YAW motion are uncoupled by roll;
2. the angular deviation between the x body and inertial axes is always equal to zero;
3. the relative axes are nonrotating and the line of sight angle is small;
4. the target has constant lateral acceleration.

While the research effectively handled the zero terminal error aspect and demonstrated improved performance, it did not consider the control constraints or the fact that the Euler angles, and consequently the control law, were changing as a result of the control inputs.

An analysis of the trajectories, in addition to the miss distance, indicated that the primary problem with the control laws using the improved Tgo was associated with the attitude dependent missile acceleration constraints that were not modeled. Consider for example the following two runs made from the same initial conditions using different guidance laws. Both of these runs are well within the launch envelope of the guidance law, use perfect position data with control inputs that are constrained by the autopilot, and are against a target that begins a level 9 "g" turn when the range equals 6000 feet. Run 1 uses G1, and Run 2 uses G4--the more sophisticated guidance law.

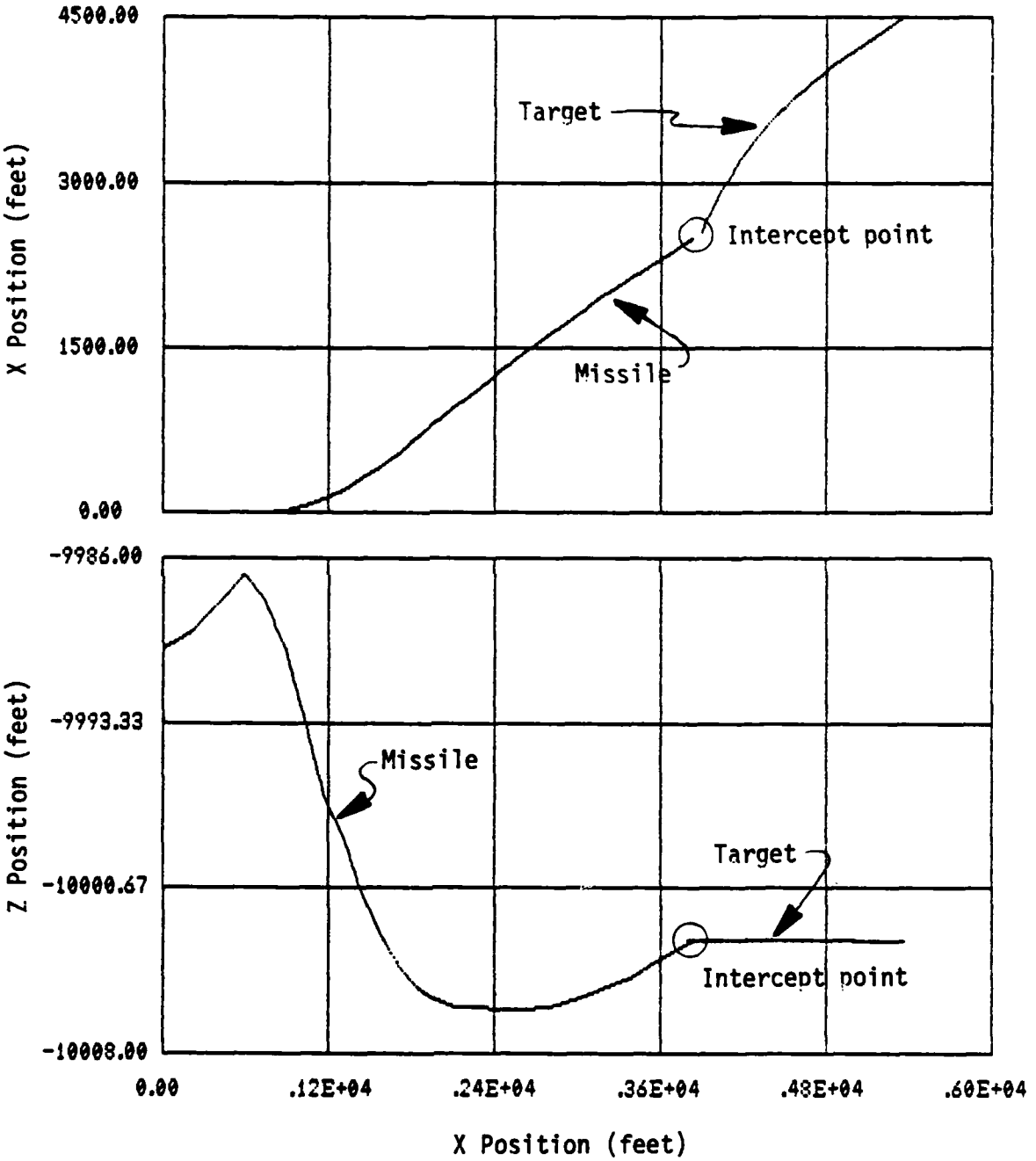


Figure 2.1A G1 CONTROLLED INTERCEPT EXAMPLE--POSITION

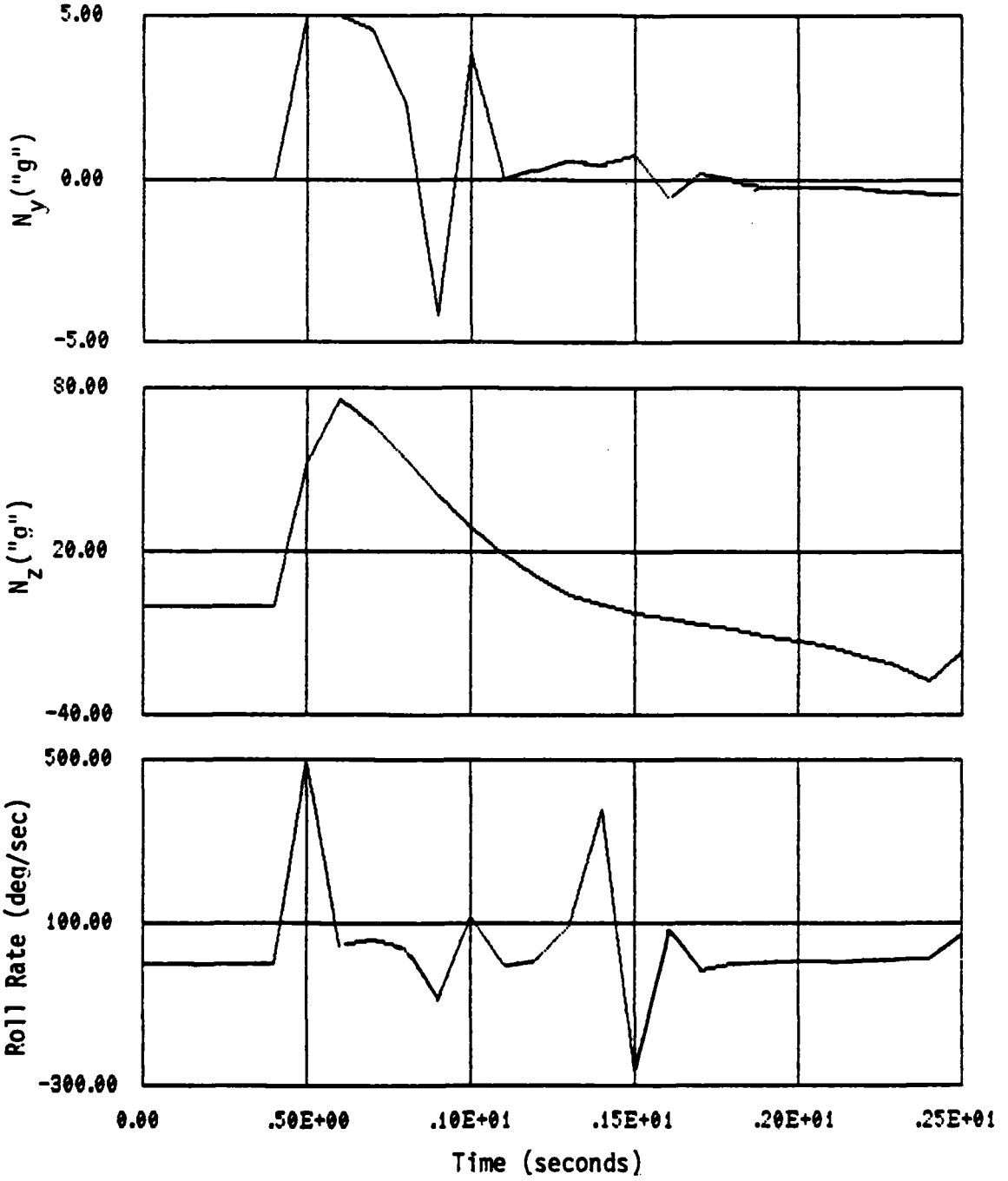


Figure 2.18 G1 CONTROLLED INTERCEPT EXAMPLE--ACCELERATIONS

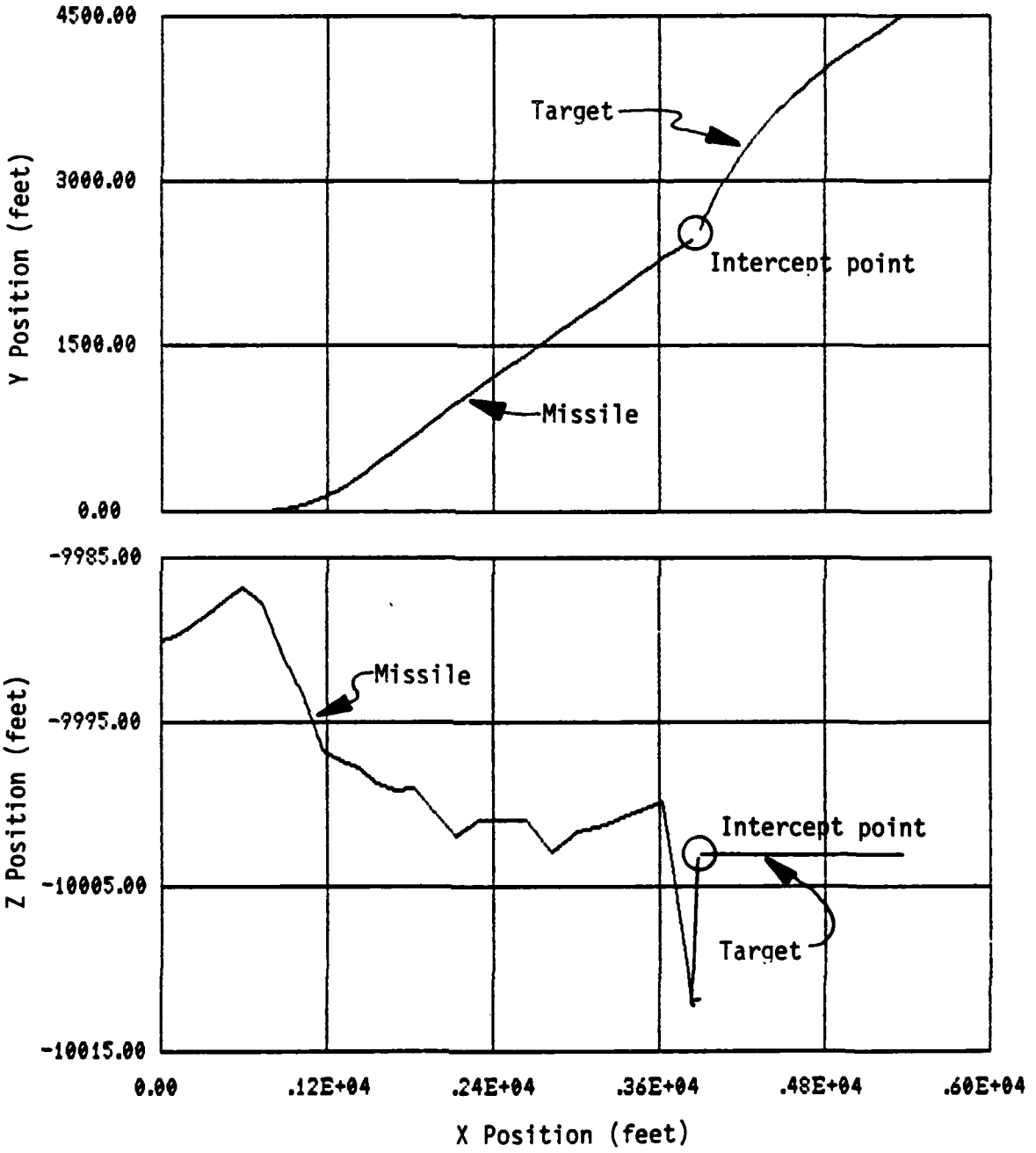


Figure 2.2A G4 CONTROLLED INTERCEPT EXAMPLE--POSITIONS

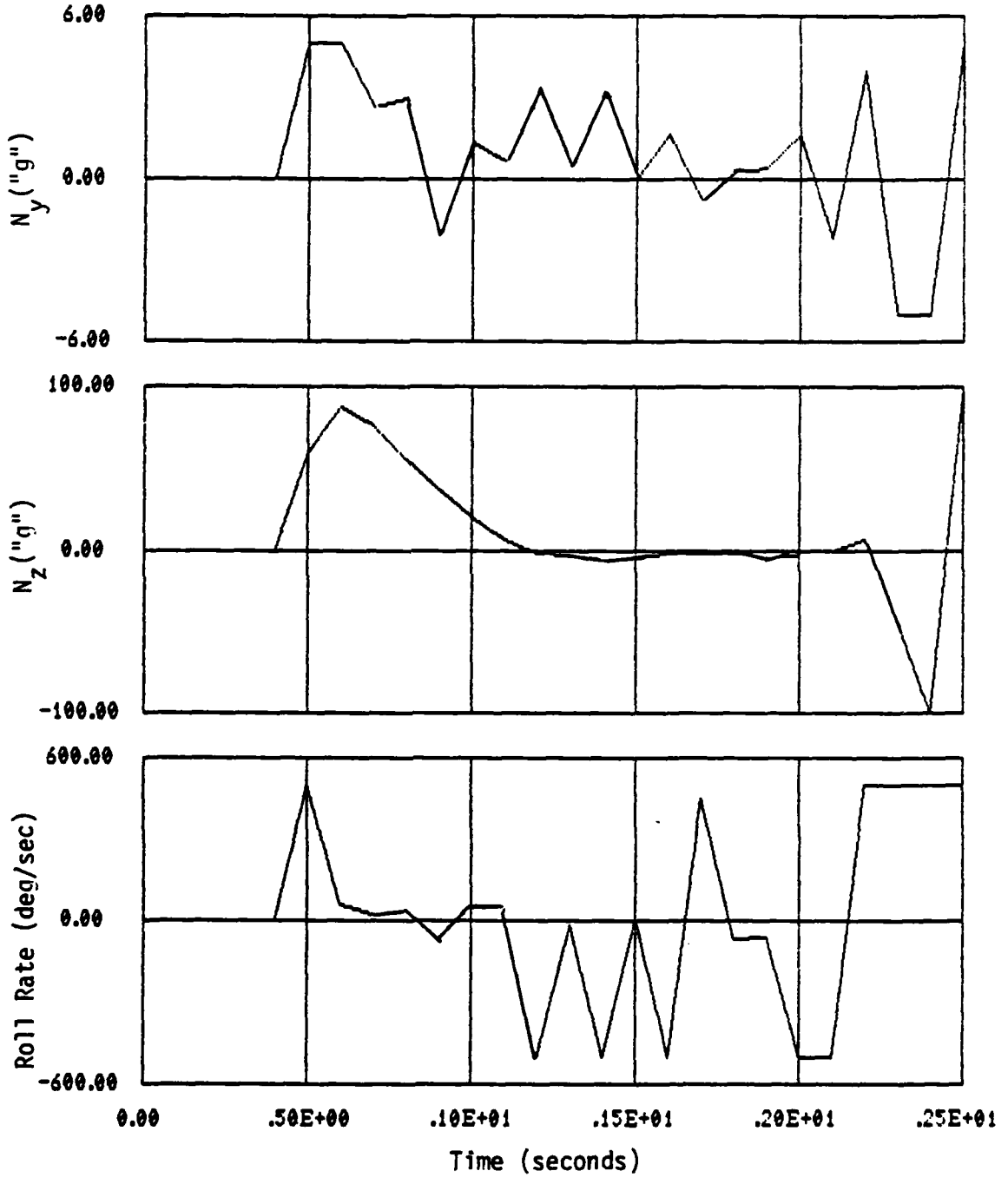


Figure 2.28 G4 CONTROLLED INTERCEPT EXAMPLE--ACCELERATIONS

As seen from Figures 2.1 and 2.2, the simpler G1 performs better than G4 in that it has a smaller miss distance (21 versus 32 feet) and more control available in the terminal phase. The difference is caused by the fact that G4 incorporates second order missile dynamics and generates a command that accounts for the imperfect autopilot. Consequently, terminal errors in the states cause higher acceleration commands as G4 accounts for autopilot delays. These commands saturate unequally and the resultant vector is rotated away from the target. This in turn causes larger deviations, further saturation and a near miss. In fact, while the acceleration commands of G1 are well within the maximum since the missile performed the roll toward the target, the G4 commands are saturated for the last .36 seconds prior to impact.

An attempt to include control constraints resulted in the nonlinear models of Fiske [8]. But these nonlinear models were determined from the same models as the linear guidance laws except that the results were passed through a saturation function. This method had two errors. First, the inertial constraints are Euler angle (attitude) dependent, and as seen from Figures 2.1 and 2.2, arbitrarily limiting one or both axes without considering the effect on the resultant acceleration command, can cause the missile to actually accelerate away from the target. Second, while the use of the saturation function seems like the natural procedure and is discussed by Tufts and Schnidman [18] and Bryson and Ho [5], the unmodified use of the saturation function may not be optimal. A discussion of the effects of constraints on control continues in Chapter III.

If the model used to develop the controller accurately describes the dynamical system and a closed form solution for this model can be

computed, then the optimal control can be based on initial solution of the unique two point boundary value problem. Consider two identical systems. If one of the processes is "interrupted" and then reinitiated using the existing values as initial conditions to the same boundary value problem, then two processes would still have the same trajectory. However, if the model used to develop the controllers was not accurate because of approximation, neglected states, etc., then the same experiment with the "simplified" controller would not produce the same trajectory, and the interrupted controller would have feedback gains that are more consistent with the accurate controller. Consequently, the "simplified" controller must be reinitialized at each guidance update [6]. In the latter case, the terminal controller is not a function of time but of only Time-to-go to the terminal state. This is an effective technique to accomodate nonlinearities, modeling errors, and uncertain target maneuvers.

In addition, the one step update gives the terminal controller with a quadratic penalty function on the terminal error some of the properties of the zero terminal error controller. In the quadratic penalty error controller, the control gains decrease to zero as the time approaches the final time. In the zero terminal error controller, the control gains grow without bound as Tgo approaches zero. When a one step update is used with the quadratic penalty error controller, the small Time-to-go couples with terminal errors resulting from the simplified model to produce acceleration commands that grow exponentially. Figure 2.3 depicts the acceleration commands from the guidance law G1, as a function of Time-to-go, for the same range and velocity. As Tgo becomes small, the acceleration commands grow exponentially, resembling the zero terminal error controller.

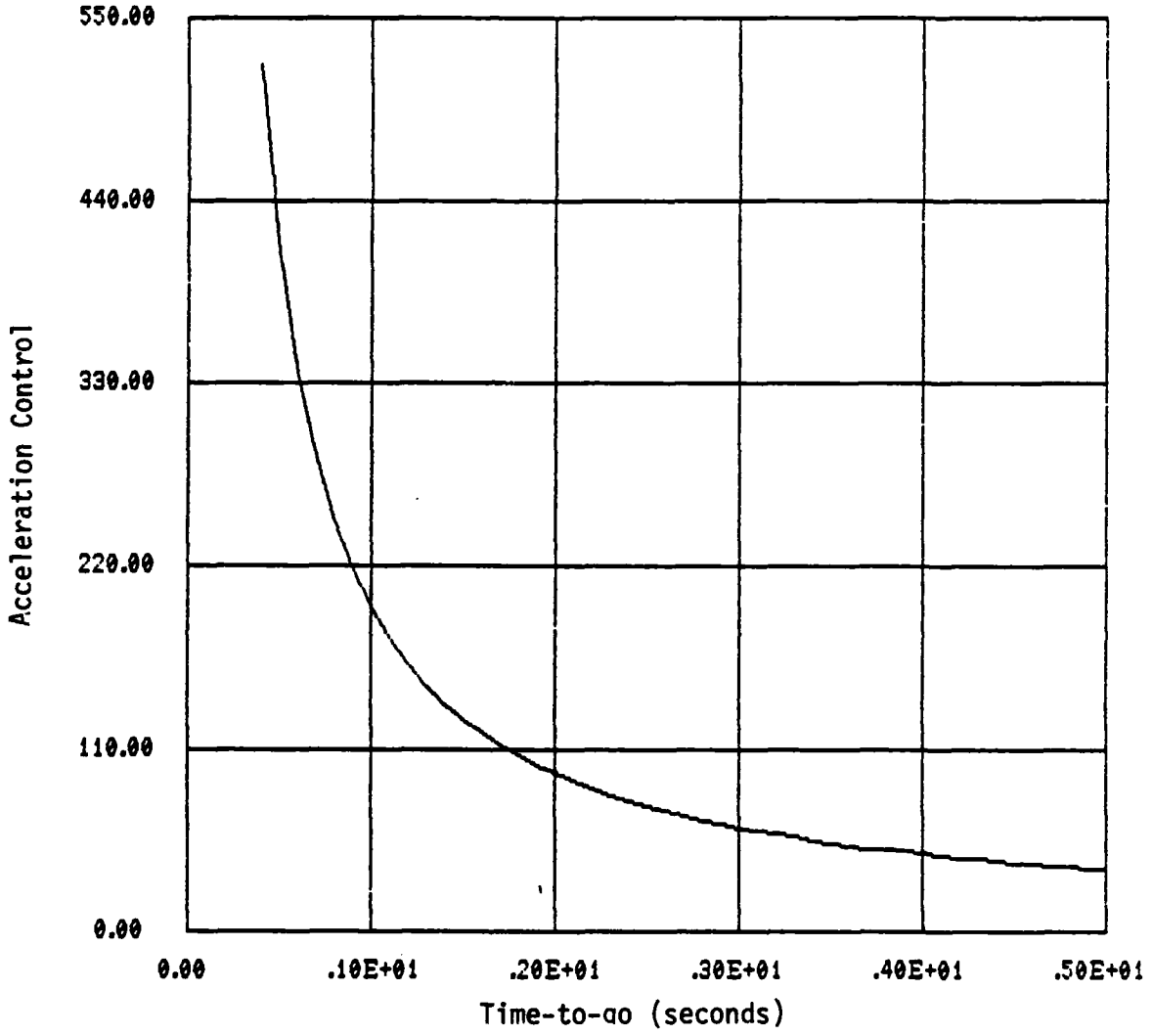


Figure 2.3 Gl Acceleration Commands

CHAPTER III
CONSTRAINED CONTROL

In a missile guidance and control system, the function of the autopilot is to realize the commanded acceleration. When the acceleration is not realizable because of aerodynamic or structural limitations, the autopilot generates the maximum allowed for each axis. In effect, the autopilot saturates each axis of the commanded acceleration independently and may significantly modify the resultant vector. Therefore, when these control constraints are introduced, the optimal control law discussed in the last chapter is no longer valid.

The optimal control law, given by Pontryagin's Maximum principle for a linear system with a quadratic performance index, is found by maximization of the Hamiltonian which is a quadratic function of the state, control and adjoint variables [5,18]. In the absence of constraints, this leads to a linear control law. With constraints, the form of the control law is still linear provided the constraint is not violated. Otherwise, the control is equal to the constraint. Due to the form of the constrained optimal control, one might assume that the constrained control can be obtained by direct application of the constraints to the unconstrained optimal control [5]. This technique, while simple to implement, may not be optimal, and often will not allow the terminal condition to be met.

In this chapter, the relationship between the constrained optimal control and the suboptimal control formed by constraining the

unconstrained optimal control will be discussed. It is shown that in certain situations (initially constrained control), when feedback controls are implemented, the two cases are identical. If the final portion of the trajectory is on the constraint, however, the system will not be able to satisfy the terminal boundary conditions unless the final condition is in the reachable set of the constrained control. Therefore, the initial (unconstrained) segment of the trajectory should be adjusted to account for the final constraint.

Because uncertainty with respect to the final state is assumed (which will also account for unmodeled errors in the dynamical system), a one-step update will be used.

Consider the state variable of the first order example and the form of the quadratic cost. The common technique in air-to-air guidance is to select relative Euclidian coordinates as the state, and control to a zero set point. This establishes the value of the state variable as the difference between the missile and target inertial positions. And consequently, the boundary condition for the relative state is always zero. Another technique better suited to solving this optimal control problem with inequality constraints on the control is to choose the seeker gimbal angles as the states, use the linear control law to determine the optimum angles, allow a non-zero set point, and solve the differential system to include the actual boundary conditions. Also, the usual quadratic cost penalizes the deviation from the final state. In this case, however, the intermediate cost is in large gimbal angles. First the gimbal angles are structurally limited. This is, in fact, a state variable constraint which is not modeled in the optimization. A larger angle increases the probability of exceeding the

gimbal angles limit when the target maneuvers. Second, because the missile has two to three times the velocity of the target, larger gimbal angles will lead to higher accelerations during the terminal phase if there are errors in the estimate of the optimal angle due to an input parameter such as relative position, velocity, acceleration, or Tgo.

Constrained Control Solution

Given: $x(t_0)$, $y(t_0)$ with $x(t_f) = y(t_f)$

Determine the optimum control such that:

$$J = \int_{t_0}^{t_f} (x^2 + u^2)\, dt \text{ is minimum}$$

subject to the constraints $|u(t)| \leq a$ and $\dot{x}(t) = u(t)$.

Writing the Hamiltonian

$$H = x^2 + u^2 + P'u$$

The adjoint is given by the solution of

$$\dot{P} = -H_x = -2x$$

and the u which minimizes H is found to be:

$$u = \begin{cases} -a \text{ sgn } (P) & |P| > 2a \\ -\frac{1}{2} P & |P| < 2a \end{cases}$$

Consider first the times when the control is unconstrained, then

$$H_u = 0 \qquad u = -\frac{p}{2} \quad \rightarrow \quad \ddot{x} = x$$

And, if $T := t_f - t_0$

$$x(t) = \left\{ \frac{y(t_f)}{e^T - e^{-T}} - \frac{x(t_0)e^{-T}}{e^T - e^{-T}} \right\} e^{(t-t_0)}$$

$$+ \left\{ \frac{x(t_0)e^T}{e^T - e^{-T}} - \frac{y(t_f)}{e^T - e^{-T}} \right\} e^{-(t-t_0)}$$

and

$$u(t) = \left\{ \frac{y(t_f)}{e^T - e^{-T}} - \frac{x(t_0)e^{-T}}{e^T - e^{-T}} \right\} e^{-(t-t_0)}$$

$$- \left\{ \frac{x(t_0)e^T}{e^T - e^{-T}} - \frac{y(t_f)}{e^T - e^{-T}} \right\} e^{-(t-t_0)}$$

Now, let $t = t_0 \rightarrow$ current time. Therefore,

$$\dot{x}(t_0) = u(t_0) = -\frac{\cosh\{ t_f - t_0\}}{\sinh\{ t_f - t_0\}} x(t_0) + \frac{y(t_f)}{\sinh\{ t_f - t_0\}}$$

which can be made exact by an integrating factor and integrated directly. The result is a closed form expression for x at any time:

$$x(t_0 + \Delta t) = \frac{\sinh\{ t_f - (t_0 + \Delta t)\}}{\sinh\{ t_f - t_o \}} x(t_0) + \frac{y(t_f)\Delta t}{\sinh\{ t_f - t_0 \}}$$

Initial Constraints

Assume that the control will be on the constraint during the initial portion of the trajectory. In this case, the saturated value is the optimal control. It is possible to calculate the time that the control will first become unconstrained. While the control law must consider the sign of the state and control, for clarity, the following equations assume a positive state, and therefore, a negative control.

While on the constraint, the state will follow:

$$x(t) = x(t_0) - at$$

Setting the expression for the control equal to "-a" and solving for "t" will provide a transcendental equation that gives the switching time from constrained to unconstrained control:

$$2y(t_f) - e^{-(t_f-t_s)}\{ a + x(t_0) - a(t_f - t_s) \}$$

$$-e^{-(t_f-t_s)}\{ x(t_0) - a - a(t_f-t_s) \} = 0$$

Here "t_0" is equal to the initial time, "t_f" the final time, and "t_s" the switching time.

Final Constraints

If the value of the optimal control is larger than the constraint during the final portion of the trajectory and the control law does not anticipate this fact, the boundary conditions will not be met. However, there may exist a Time-to-go and state such that the subsequent use of the saturated control will cause the system state to just satisfy

boundary conditions. If the control is equal to the saturated value "a" and the state is equal to:

$$x(t_s) = y(t_f) - a(t_f - t_s),$$

then the terminal boundary conditions will be met.

Because the closed loop control is not defined at $t = t_f$, and the control law only provides the fact that

$$\dot{x} = u$$

the open loop control expression must be used to determine this time.

Solving for time in the open loop equation with:

$$y(t_s) = y(t_f) - a(t_f - t_s)$$

$$u(t_s) = a$$

$$t_f = t_s$$

$$T = t_s - t_0$$

$$u(t) = \left\{ \frac{y(t_s)}{e^{T} - e^{-T}} - \frac{x(t_0)e^{-T}}{e^{T} - e^{-T}} \right\} e^{(t_s - t_0)}$$

$$- \left\{ \frac{x(t_0)e^{T}}{e^{T} - e^{-T}} - \frac{y(t_s)}{e^{T} - e^{-T}} \right\} e^{-(t_s - t_0)}$$

The result is the following equation for t_s:

$$a \sinh\{t_s - t_0\} - \{ y(t_f) - a(t_f - t_s) \} \cosh\{t_s - t_0\} + x(t_0) = 0$$

This equation can be solved for the associated switching time. Then, this time "t_s", and boundary condition--

$$x(t_s) = y(t_f) - a(t_f - t_s)$$

are used in the solution of an unconstrained optimal control problem. The result will be a control that meets the actual boundary condition "$y(t_f)$" at "t_f".

Examples

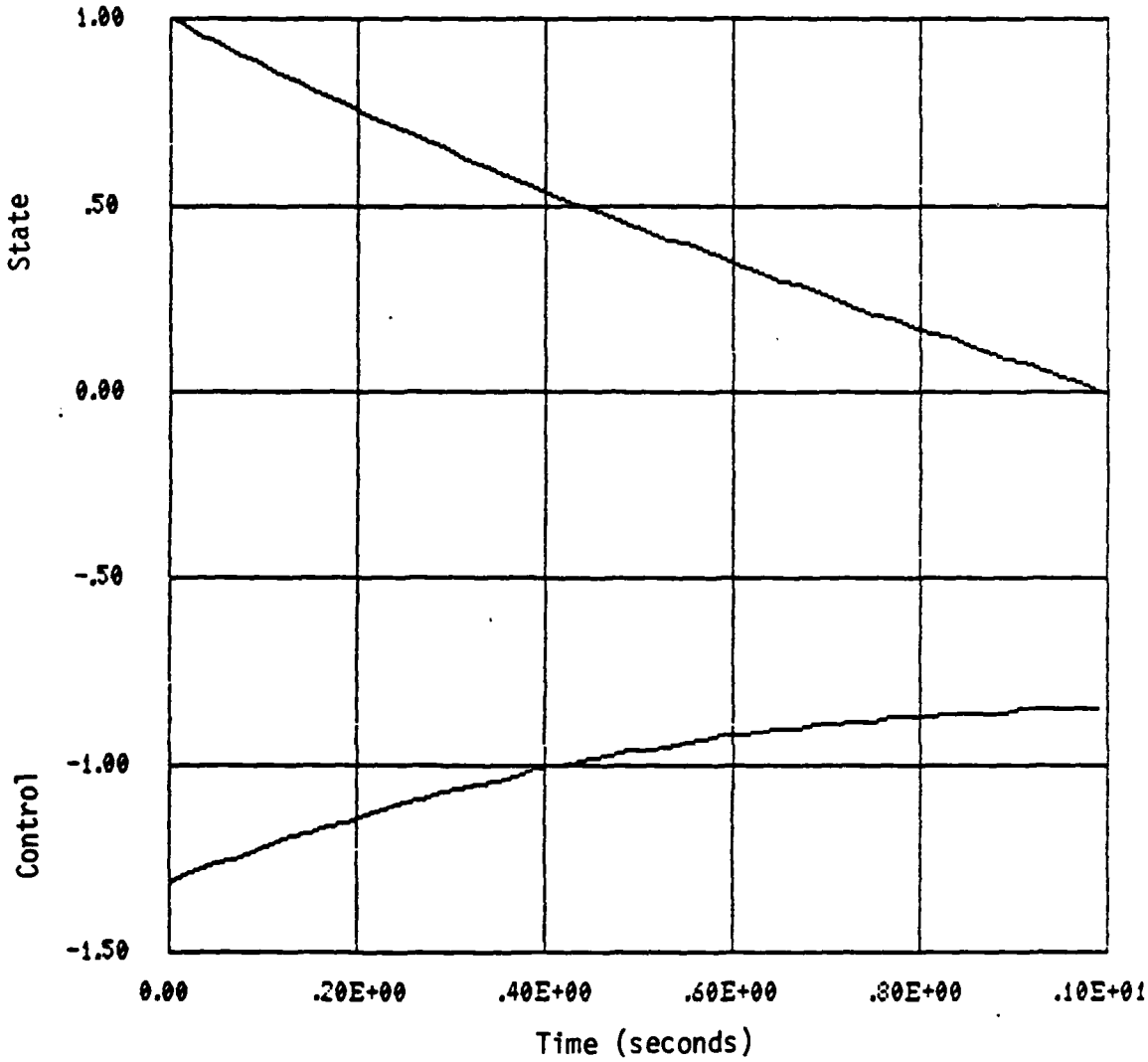
The following figures illustrate the use of the computed switching times to account for boundary conditions. In these first order examples, the initial value of the state is 1, the initial boundary condition is the zero state, and the control is constrained to a value of 1.1. If the terminal boundary condition changes, it does so at .5 seconds and is set again at 1. Also, in these examples, the following definitions are used:

Unconstrained = There are no contraints on the control magnitudes.

Suboptimal = There are constraints on control magnitudes, but an unconstrained control is calculated and passed through the saturation function.

Constrained = There are constraints on control magnitudes, and the control is calculated to account for the constraint.

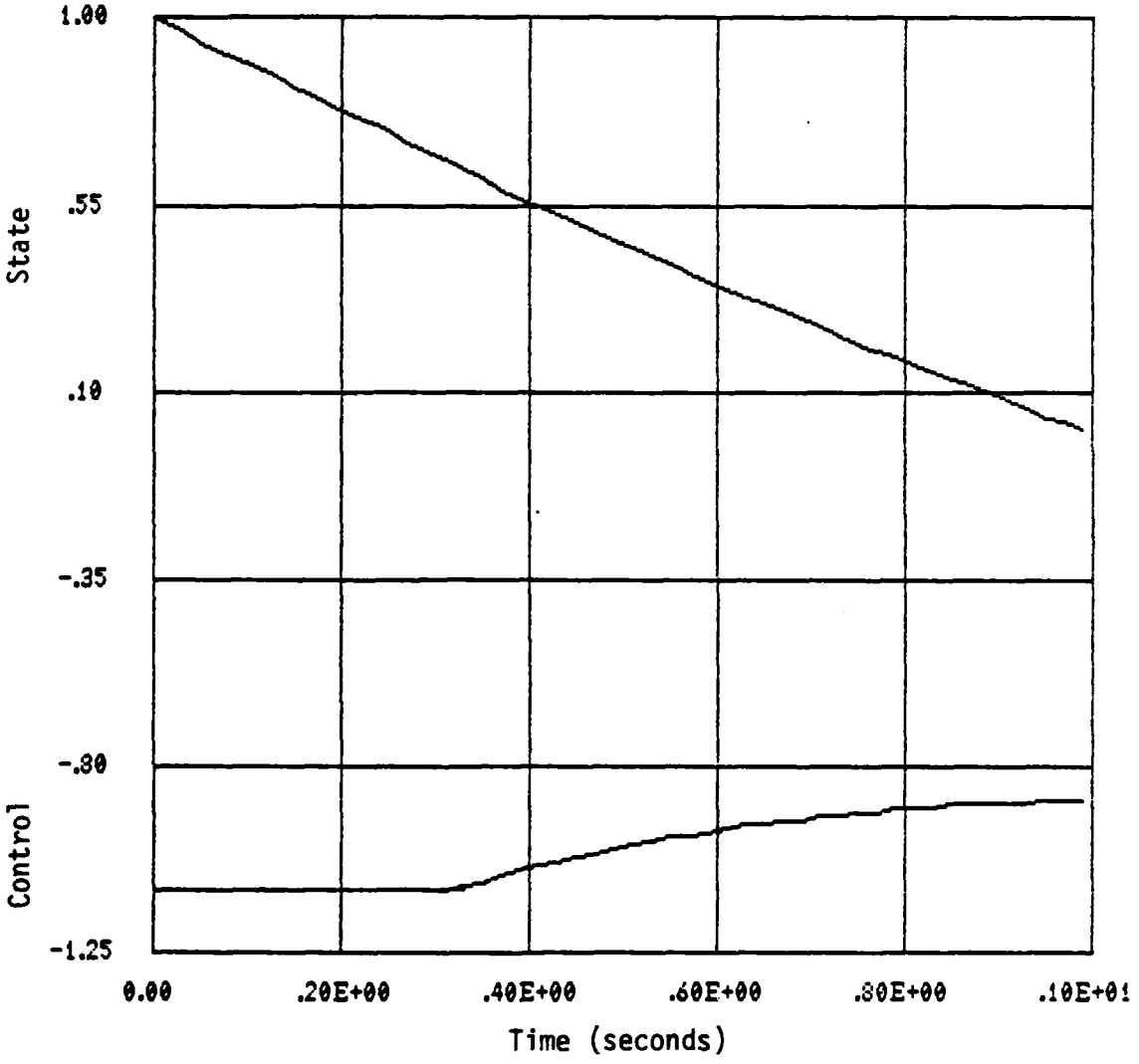
Reachable Set = There are constraints on control magnitudes, and the control is calculated to ensure that the final control is off of the constraint.



Final State Value = 0

Figure 3.1 UNCONSTRAINED CONTROL--SINGLE FINAL CONDITION

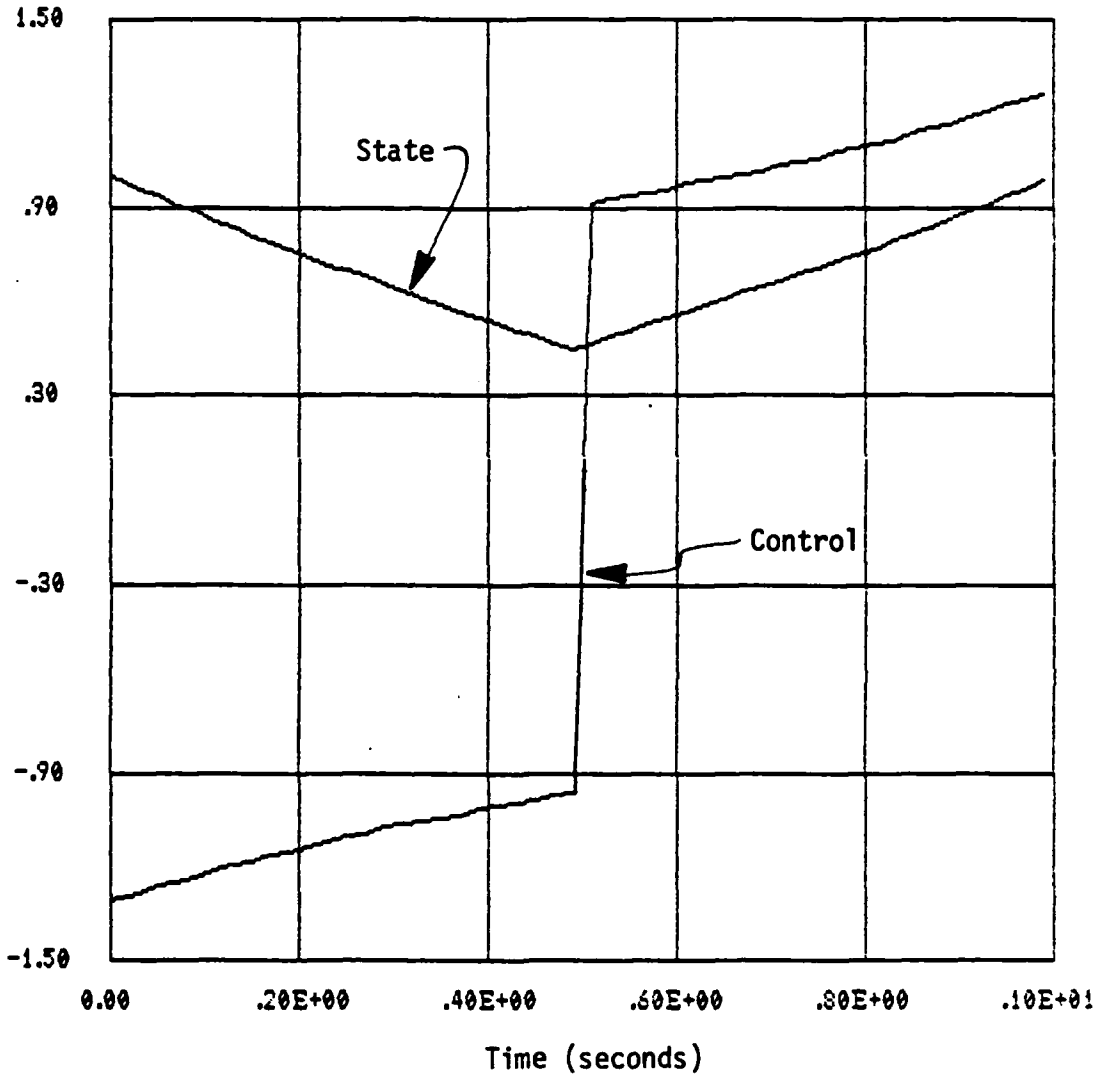
In Figure 3.1, the unconstrained control and state values are presented. There is only one final condition.



Final State Value = 0

Figure 3.2 SUBOPTIMAL CONTROL--SINGLE FINAL CONDITION

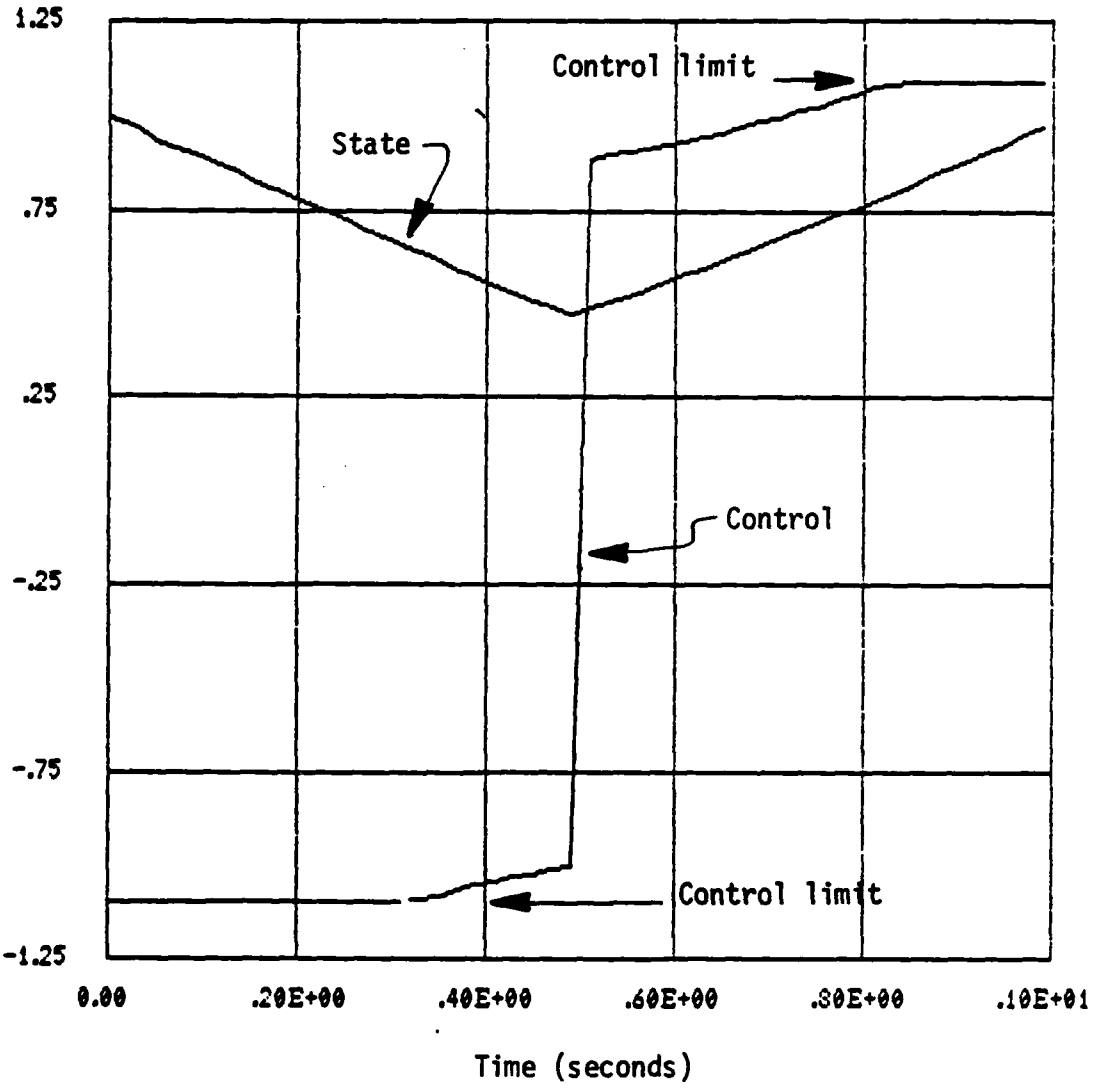
Figure 3.2 shows the effect of an initial constraint.



Final State Value = 1.

Figure 3.3 UNCONSTRAINED CONTROL--MULTIPLE FINAL CONDITIONS

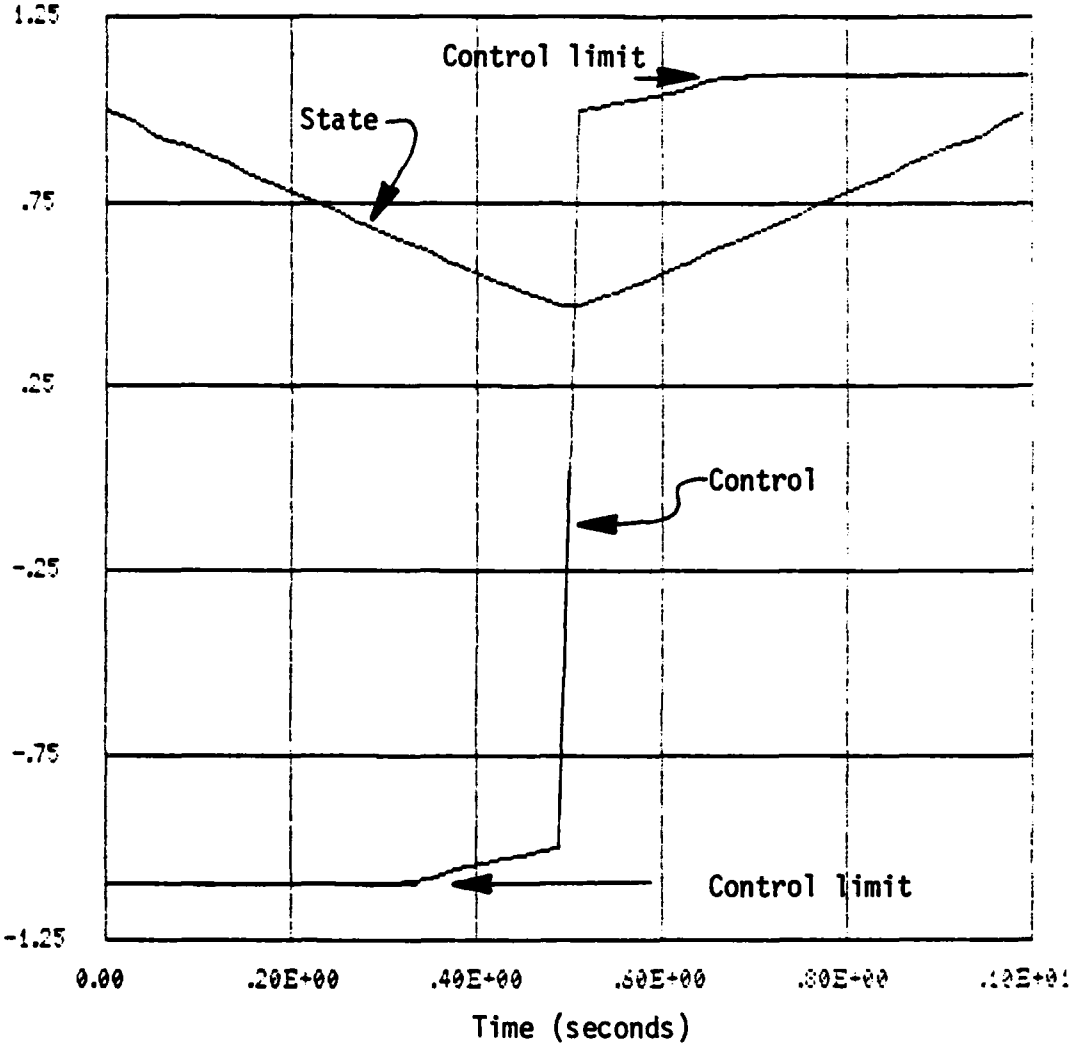
Figure 3.3 shows the system behavior when the terminal condition is changed at t = .5. Again, there are no constraints on the control.



Final State Value = .975

Figure 3.4 SUBOPTIMAL CONTROL--MULTIPLE FINAL CONDITIONS

In Figure 3.4, the effect of the unanticipated final constraint is illustrated. In this case, the final boundary condition is not met.



Final State Value = 1.0

Figure 3.5 CONSTRAINED CONTROL--MULTIPLE FINAL CONDITIONS

Figure 3.5 shows the proper handling of the final constraint. In this case, the unconstrained terminal conditions are adjusted so that the final condition is met with the saturated control.

While the constrained control in Figure 3.5 does meet the boundary conditions, the terminal condition is on the boundary of the reachable set. Any additional change after the control is saturated (in the direction away from the origin) cannot be controlled by the input. If the control is going to saturate during the final phase, another technique is shown in Figure 3.6. Here the control is placed on the constraint until the switching time when unconstrained control is allowed. This procedure will ensure that the control is in the linear region for the terminal portion of the intercept and is a partial alternative to the reachable state control algorithm presented in [6].

Summary

In summary, to optimally handle constraints two cases must be considered. In the first case, a saturated control unsaturates prior to the final time. Here, the use of the saturation function is the proper control. In the other case, however, the control is such that it saturates during the terminal stages of the intercept. Now, the fact that the control is going to saturate must be considered. In the case of the first order trajectory, this is handled by changing the final time and boundary condition of the unconstrained portion of the control. Reversing the time that the saturated and unconstrained control is applied ensures that the target remains well within the reachable set of the missile control.

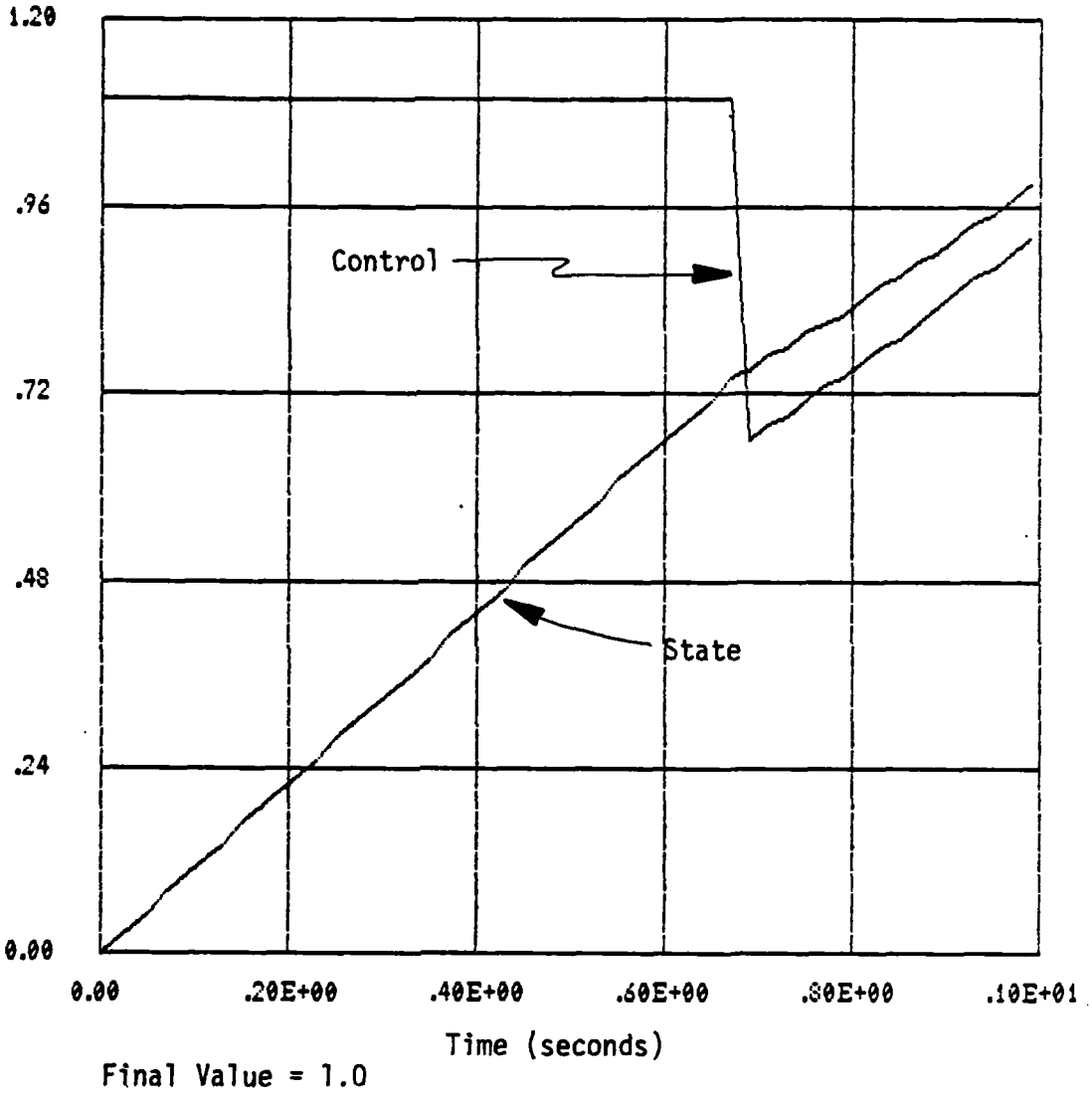


Figure 3.6 REACHABLE SET CONTROL

CHAPTER IV
NONLINEAR CONTROL LAW

Numerous attempts were made to incorporate the attitude dependence of the engagement in a state space model by including the Euler angles in the model formulation and exploiting the properties of the bank-to-turn missile. If these angles could be included, the control laws would be optimal in the dynamic-programming sense because the same optimal control laws would be used in the future [17]. Considering that the control inputs cause the change in the inertial and relative attitudes and that the Euler angles describing the missile attitude and the seeker angles describing the relative angular differences can be large, the system matrices are time and control dependent. Consequently, the model is nonstationary and the limiting solution of the algebraic Riccati equation is not appropriate. Working directly with the Euler-Lagrange equations, approximate methods such as singular perturbations and the method of averaging were attempted [19,20,21]. But even with approximations, the result was a coupled set of nonlinear equations, and the major value of the approximation methods was lost [22]. Consequently, each effort was terminated when the assumptions and/or approximations required for a closed form solution invalidated the purpose of incorporating the increased complexity. The only result of these tedious exercises was the author's agreement with Gupta et al. [6]: "When the detailed translational and rotational missile model is used, the resulting TPBVP [Two Point Boundary Value Problem] is so complex

that it cannot be solved in real time even with the most advanced digital hardware".

Instead, since the major problem with the current guidance laws was the unmodeled constraints, a scheme was developed that would incorporate the optimal guidance laws and still apply the acceleration commands consistent with the control constraints.

Guidance Law

The seeker gimbal angles provide a direct indication of the target-missile geometry. Using the seeker-to-missile body transformation outlined in Appendix B and the dynamics approximations presented in Appendix C, the following relationships can be derived:

$$\omega_y = -p \sin \phi_g - (N_z/u) \cos \phi_g$$

$$\omega_z = p \sin \theta_g \cos \phi_g - (N_z/u) \sin \theta_g \sin \phi_g + (N_y/u) \cos \theta_g$$

where ω_y = missile rotation about the seeker y axis

ω_z = missile rotation about the seeker z axis

θ_g = seeker elevation angle

ϕ_g = seeker azimuth angle

N_y = acceleration along the y body axis

N_z = acceleration along the z body axis

u = missile velocity along the x body axis

p = missile roll rate about the x body axis

These equations provide a method of generating the linear accelerations to control the missile's relative attitude by controlling the gimbal angles since the change in the seeker angles due to missile attitude is

ω_y and ω_z. A positive rotation about either axis causes a negative change in the seeker angle. Therefore:

$$\dot{\theta}_g = -\omega_y = +p \sin \phi_g + (N_z/u) \cos \phi_g$$

$$\dot{\phi}_g = -\omega_z = -p \sin \theta_g \cos \phi_g + (N_z/u) \sin \theta_g \sin \phi_g - (N_y/u) \cos \theta_g$$

Consequently, an "attitude controller" can be developed using the first order control laws from Chapter III and steering from the advanced guidance laws. The inertial controls are used to generate the "optimal" trajectory (not accounting for constraints). Then these relative directions are rotated to the missile body axis and compared to the existing seeker angles to determine the optimal seeker angles. These optimal angles are used as boundary conditions for a constrained control law that relates the angles to the control inputs.

Longitudinal Control

Setting the state equal to the seeker angle θ_g, the control input to $\dot{\theta}_g$, and considering only longitudinal motion ($p = N_y = 0$) the ω_y relationship leads to the following control:

$$N_z = -\left\{\frac{(e^T + e^{-T})\theta_g}{e^T - e^{-T}} - \frac{2\theta_{opt}}{e^T - e^{-T}}\right\} \frac{u}{\cos \phi_g}$$

where θ_{opt} = the θ_g derived from the linear control law.

Lateral-Directional Control

N_y. Setting the state equal to the seeker angle ϕ_g, the control input to $\dot{\phi}_g$, and solving for N_y with p and N_z equal to zero:

$$N_y = +\left\{\frac{(e^T + e^{-T})\phi_g}{e^T - e^{-T}} - \frac{2\phi_{opt}}{e^T - e^{-T}}\right\} \frac{u}{\cos \theta_g}$$

ROLL. While the gimbal angle ϕ_g is controlled by the N_y acceleration, the magnitude of this control is severely limited. Therefore, to handle this constraint, the control law must be expanded to position the missile such that the N_z acceleration can be used to achieve the intercept. Since the gimbal angle ϕ_g is equal to the roll angle difference between the maximum maneuver plane of the missile and the relative target position, the azimuth angle can be used in place of the roll angle difference to generate the roll rate command. Once the roll rate command is incorporated, however, the approximation used in the N_y equation is not always valid. When there is a large roll angle difference, p will be the dominant control with a small contribution from N_y (which will also help control adverse yaw on a higher order system). As the roll angle difference approaches zero, the roll rate will decrease while N_y continues to control the gimbal angle ϕ_g during the terminal phase.

$$p = + \left\{ \frac{(e^{T} + e^{-T})\,\phi_g}{e^{T} - e^{-T}} - \frac{2\,\phi_{opt}}{e^{T} - e^{-T}} \right\}$$

Time-to-Go

Even though Rigg's algorithm improved the performance of the advanced guidance laws, it continued to underestimate the time-to-go during the initial segment of the trajectory. This is because the algorithm neglected the attitude of the missile when setting the X inertial acceleration to the actual acceleration (x body axis) of the missile. Rigg's algorithm is repeated here:

$$Tgo = \frac{2R}{Vc + SQRT(Vc^2 + 4AR)}$$

where Tgo = time-to-go

T_b = time during which the missile is accelerating

t = time

R = range

Vc = closing velocity

A = average acceleration

Amax = acceleration during boost

Amin = deceleration after burnout

As shown by Lee [23], the actual time-to-go is a function of the trajectory (including constraints). While the constraints cannot be easily handled, the difference between the inertial and body axis accelerations can be corrected by considering the angular differences.

As seen from the Tgo formula, in addition to the estimate of missile acceleration, the closing velocity (Vc) is used. This term is substituted for the X component of relative velocity in the derivation. Vc, however, is along a vector from the missile to the target and not in the X direction. If the Euler angles that describe the missile's orientation are included in resolving the difference between the X acceleration command and the missile orientation, the result is a cubic equation containing the direction cosines, inertial positions, velocities, and accelerations.

A simpler technique is to correct the closing velocity, and resolve the component in the x missile direction. This is accomplished by calculating the spherical angle (Aoff) between the missile x axis and target and then using the x component of the closing velocity in the computation of the x commanded acceleration.

$$Aoff = \tan^{-1} \left\{ \frac{\sqrt{(z^2 + y^2)}}{x} \right\}$$

$$A = A \cos (Aoff)$$

where x = relative x distance to the target in the body axis frame.

y = relative y distance to the target in the body axis frame.

z = relative z distance to the target in the body axis frame.

Another minor correction was made to the Tgo algorithm by correcting the estimate of average acceleration which is A above. The algorithm calculates this value and then, if the time is greater than the motor burn time, the acceleration estimate is set to Amin. However, if the time-to-go is less than the missile burn time, then the second component, the deceleration term, is added to the acceleration term in the computation of average acceleration.

A better acceleration estimate is calculated if all three cases are considered separately:

1. intercept during the boost phase, time less than T_b;

$$A = \frac{Amax\ (T_b - t)}{Tgo}$$

2. intercept during the coast phase, time less than T_b;

$$A = \frac{Amax(T_b - t) + Amin(Tgo + t - T_b)}{Tgo}$$

3. current time greater than accereration time;

$$A = Amin$$

CHAPTER V
RESULTS

Four topics have been discussed with respect to the control of the bank-to-turn missile. These are the effect of constraints on control, the linear optimal control laws, the improvement of the Time-to-go estimate, and a nonlinear control law. The results of additional evaluations, the modifications incorporated, and the nonlinear control law will be presented here.

Constrained Control

The four trajectories used to evaluate the effect of constraints are shown in Table 5.1:

Table 5.1 Boundary Conditions

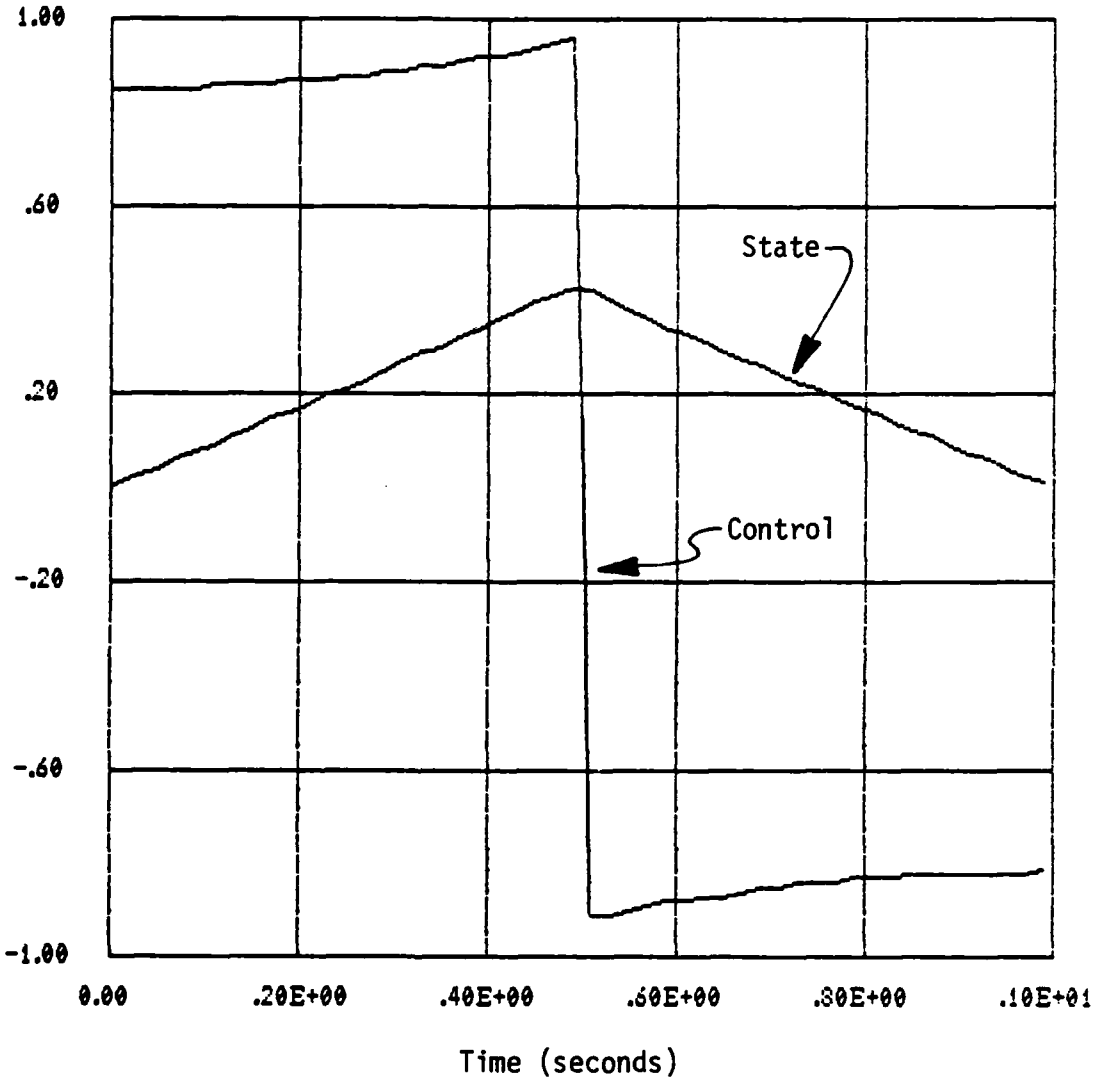
Trajectory	Initial Conditions	Terminal Condition Time	Terminal Condition Value
1	1	0	0
2	0	0	1
3	1	0	0
		.5	1
4	0	0	1
		.5	0

Table 5.2 presents the cost for each of the trajectories. Here the same terminology is used that was introduced in Chapter III. As seen from the table, the optimal control is the minimum cost solution that meets the final value when used with the trajectories that have a single boundary condition (trajectories 1 and 2). Also as expected, a control constraint increases the cost, even if it is handled in an optimal fashion, if there is no change in the terminal boundary condition.

Table 5.2 Cost Function

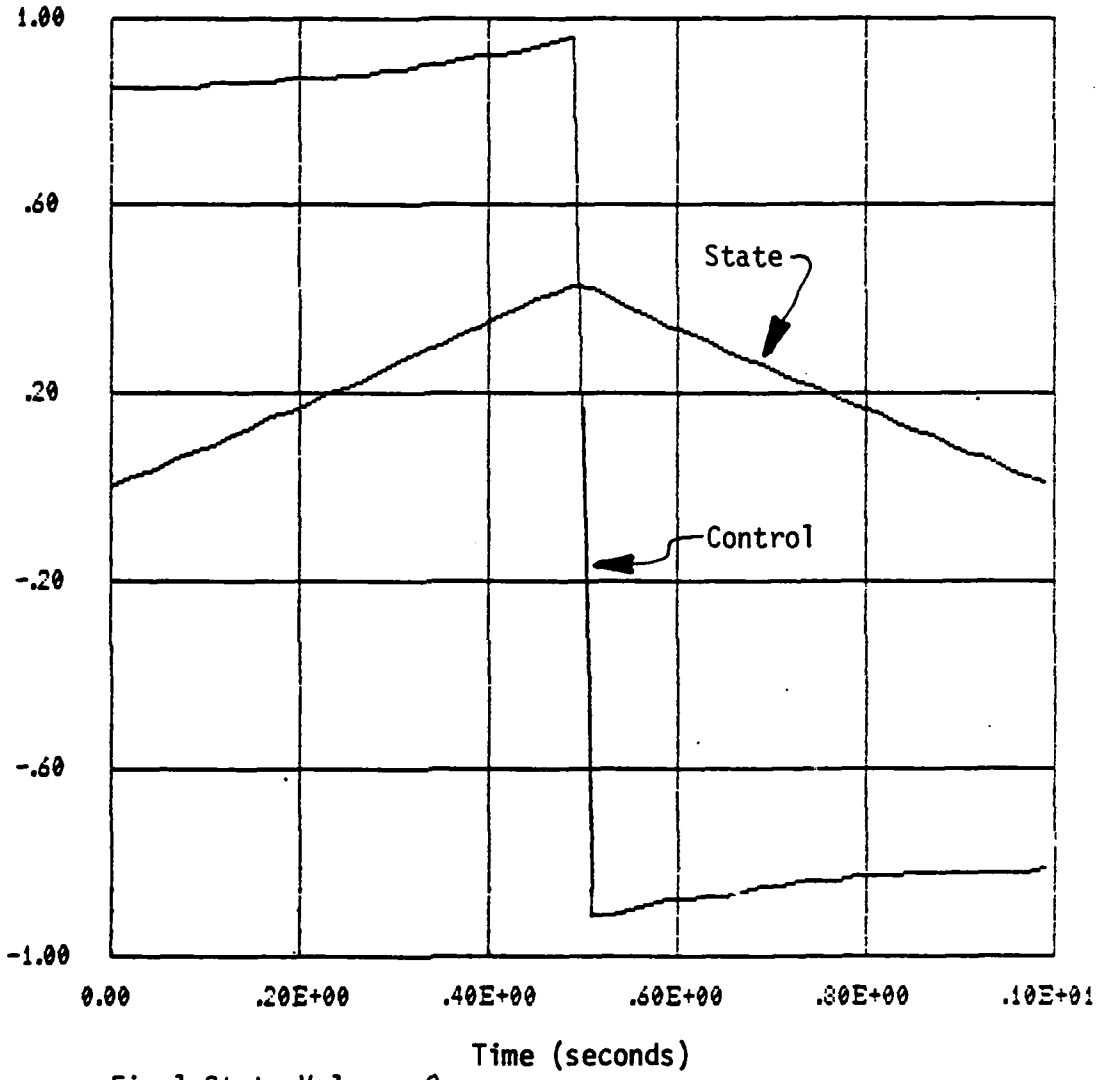
Trajectory	Control	Cost	Final Condition
1	Unconstrained	1.323	Yes
	Suboptimal	1.325	Yes
	Constrained	1.325	Yes
	Reachable Set	1.325	Yes
2	Unconstrained	1.303	Yes
	Suboptimal	1.243	No
	Constrained	1.311	Yes
	Reachable Set	1.392	Yes
3	Unconstrained	1.740	Yes
	Suboptimal	1.654	No
	Constrained	1.718	Yes
	Reachable Set	1.722	Yes
4	Unconstrained	.816	Yes
	Suboptimal	.816	Yes
	Constrained	.917	Yes
	Reachable Set	1.312	Yes

The trajectories with changing terminal conditions, where the constraint is never violated, however, display some unexpected properties and have been included as Figures 5.1 through 5.3. First, the optimal control may not lead to the lowest cost. If the control expects to be on the constraint during the final portion of the



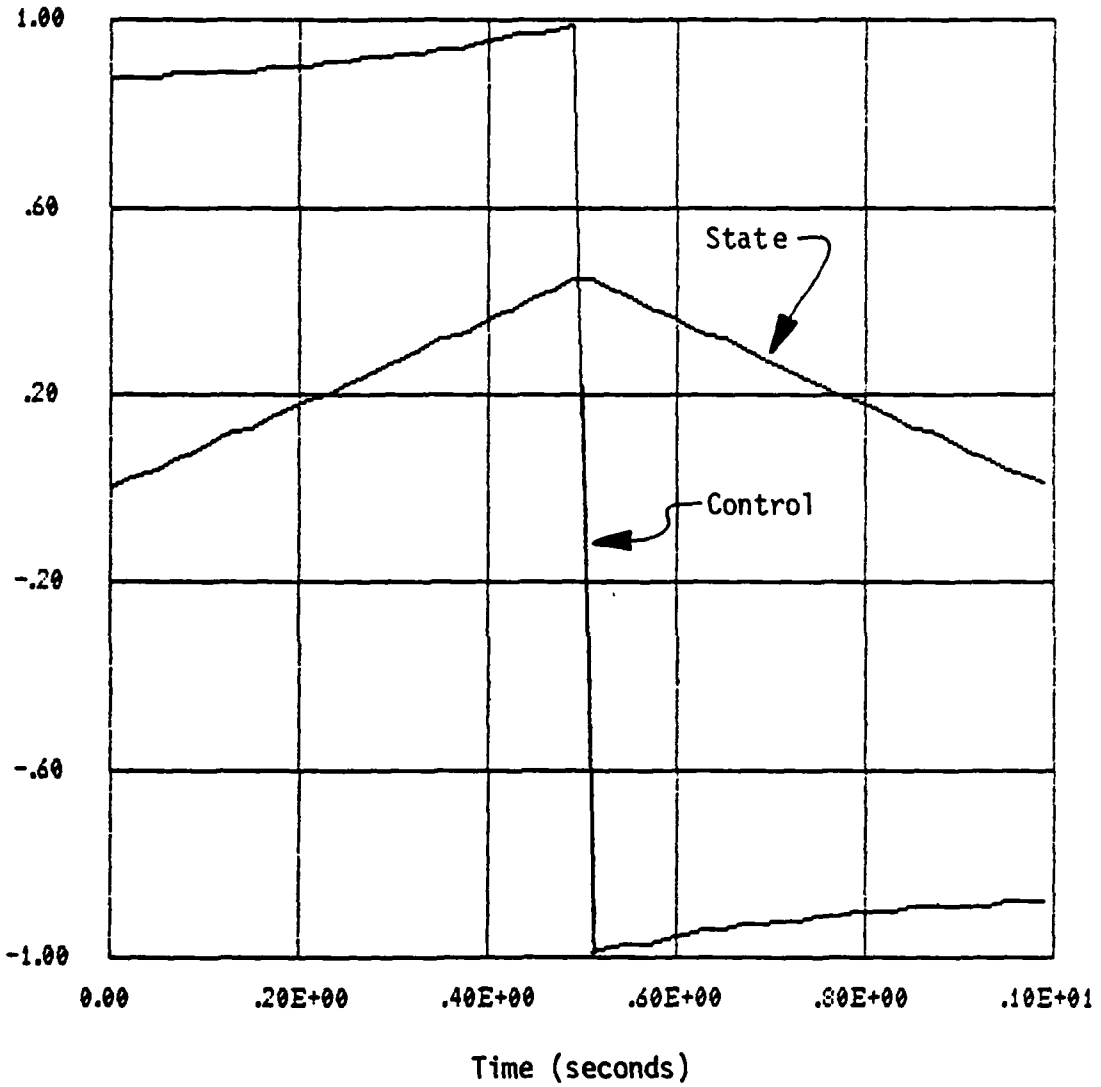
Final State Value = 0

Figure 5.1 UNCONSTRAINED CONTROL



Final State Value = 0

Figure 5.2 SUBOPTIMAL CONTROL--WITH UNREALIZED CONSTRAINTS



Final State Value = 0

Figure 5.3 CONSTRAINED CONTROL--WITH UNREALIZED CONSTRAINTS

trajectory, it increases the control during the initial phase to insure that the final condition is met. If the final condition is changed, then the increased control may not have been required. In fact, the increased control can result in a larger displacement from the new terminal condition. The result is an overall higher cost for the complete trajectory.

Second, with a discontinuous boundary condition, the constraint may not lead to an increase in cost. Again, this is a function of the initial conditions and the change in the final condition. The explanation for this effect is similar to the first in that the constraint may keep the state closer to the second terminal condition.

While these are interesting observations, it should be noted that the changes in the terminal boundary condition are quite severe and that these effects are dependent on the changes and when they occur. Also, if the boundary condition is within the reachable set, the only control that can always meet the condition is the properly formulated optimal control.

Linear Optimal Control

Generally, the performance of a guidance law for air-to-air missiles is measured by the inner and outer launch boundaries. These are defined as the minimum and maximum range from which the missile can be launched and achieve a hit. The boundaries are a function of the initial conditions of the engagement such as speed, altitude, relative attitude, and geometry. The geometry of the engagement is defined by the off-boresight angle and the aspect angle. Figure 5.4 depicts these angles. The off-boresight angles (OBA) are defined as the angles

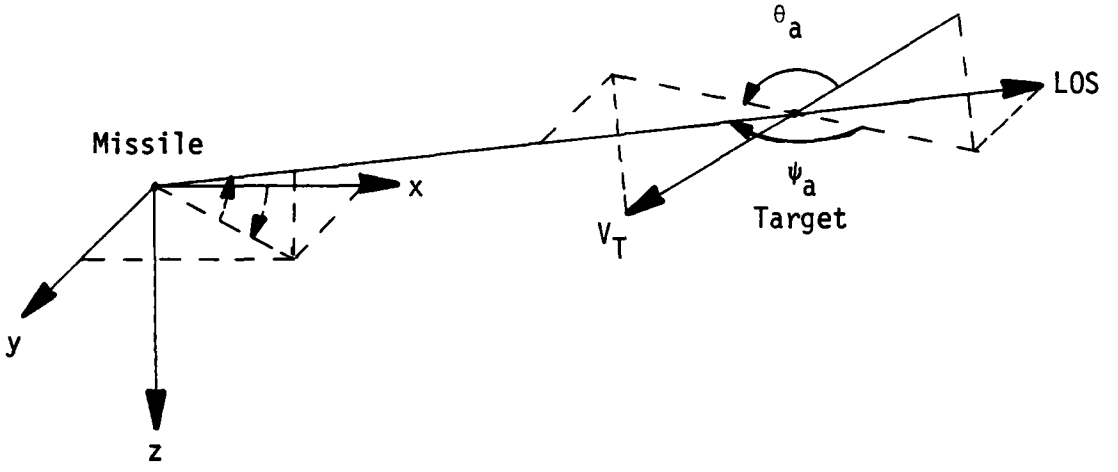


Figure 5.4 ENGAGEMENT GEOMETRY

(azimuth and elevation) between the missile velocity vector and the LOS. The aspect angles are defined as the angles, measured from the tail of the target, between the target velocity and the missile LOS. In all of the cases studied, the target and missile are co-speed (.7 Mach) and approximately co-altitude (10,000 feet) at launch.

Previous work [1,9] with the linear optimal control presented by Fiske [8] utilized an "evasive" target maneuver developed by the Air Force Armament Laboratory. This maneuver has two phases. For a target aspect angle less than 90 degrees (tail shot) at a range of 6000 feet, the target accelerated into the missile at an angle of 45 degrees (down) with respect to the vertical. For a target aspect angle greater than 90 degrees, the target also rolls into the missile, except that this time the angle is 45 degrees up with respect to the vertical. In each case, at one second-to-go the target rolls to 180 degrees and accelerates down (positive Z direction). Unfortunately, this is not a very evasive maneuver. To begin with, by automatically rolling into the missile the

maneuver actually reduces rather than increases the angular travel, the line of sight rates, and the average acceleration required. Also, for the short range shots, the fact that the target is maneuvering at a 45 degree angle while the missile is waiting for the clearance delay (.4 seconds) reduces the amount of ROLL required to place the target in the primary maneuver plane. Recalling the limitations of the missile and the formulation of the guidance laws, much simpler and more effective maneuvers can be devised.

First, consider the implementation of the target used for this evaluation of the Linear Optimal Control Laws. The target is a three (3) dimensional, nine (9) "g" maneuvering target. In order to prevent the computational burden of integrating target accelerations for velocity and position, the target is mechanized by computing the radius of curvature and the angular velocity of a constantly accelerating point and incorporating the equations as uniform circular motion. The plane of maneuver for the target is determined by the target ROLL angle, an input parameter. When the maneuver is initiated, the center of rotation, radius of curvature, and the coordinate transformations are computed. Then, the inertial positions, velocities, and accelerations are computed by transforming the maneuver plane motion, a simple function of the angle of rotation referenced to the start of the maneuver, to the inertial reference system.

Using this target, a deterministic evaluation of the inner launch boundary was initiated using an improved six degree of freedom simulation based on the simulation used in [1]. It soon became obvious that the concept of inner and outer launch boundaries for these missile/guidance law combinations did not apply. The launch boundaries

are supposed to define an envelope within which the missile will intercept the target. This is not the case with the bank-to-turn missile and linear guidance laws. To illustrate this point, consider the trajectory shown in Figure 5.5. The launch range is 5000 feet, the OBA is 40 degrees, and the aspect angle is 180 degrees. At launch, the target begins a level 9 "g" turn away from the missile. Midway through the flight, the guidance law essentially has the intercept conditions met and reduces the acceleration commands. Toward the end of the attempted intercept, however, increased accelerations are required, the missile control saturates, and the missile misses a relatively simple intercept. The miss was not caused by a lack of maneuver capability, the basis for the launch envelope, but by neglected attitude dependent constraints.

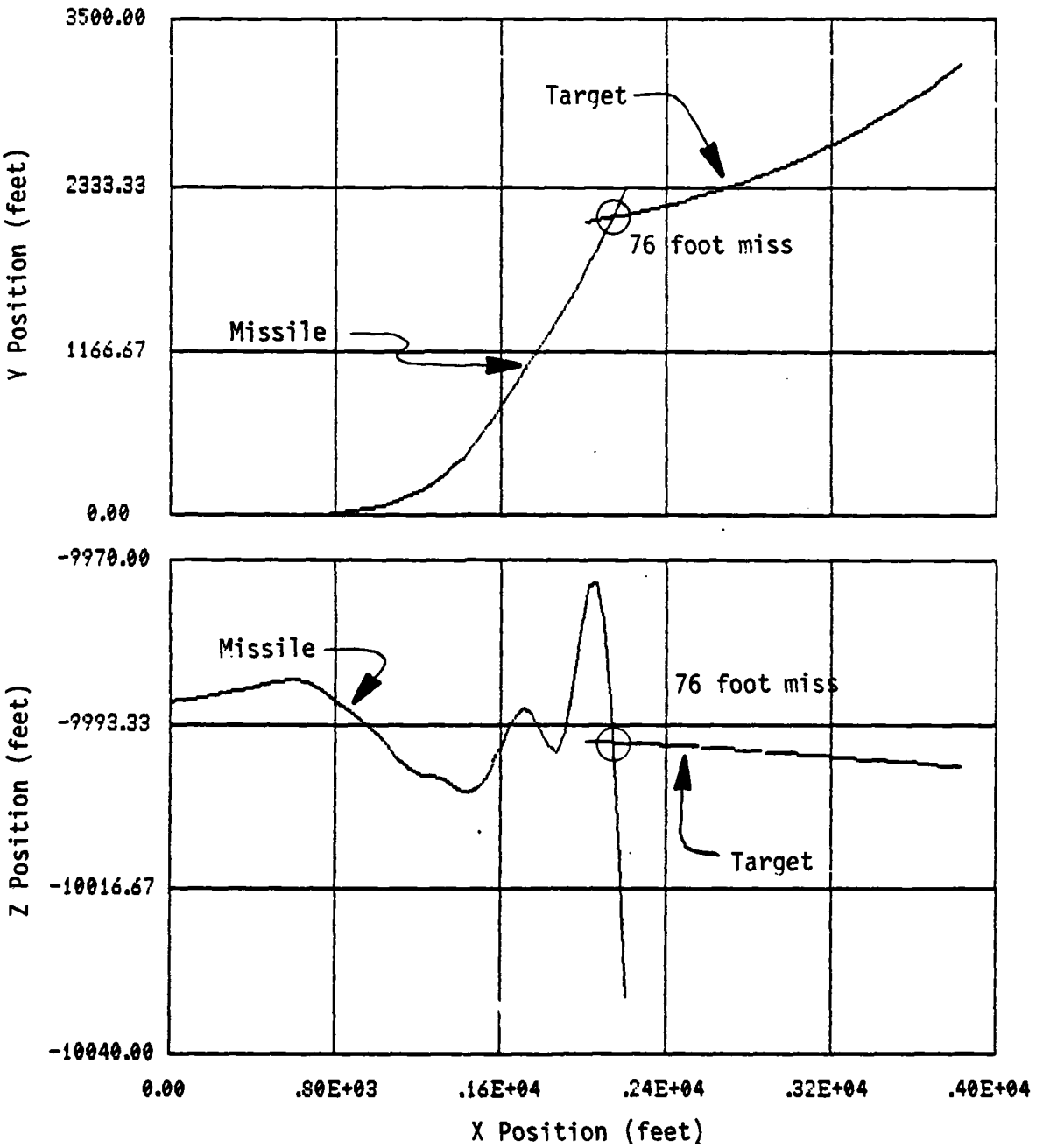


Figure 5.5A G4 MISSED INTERCEPTS--TRAJECTORY

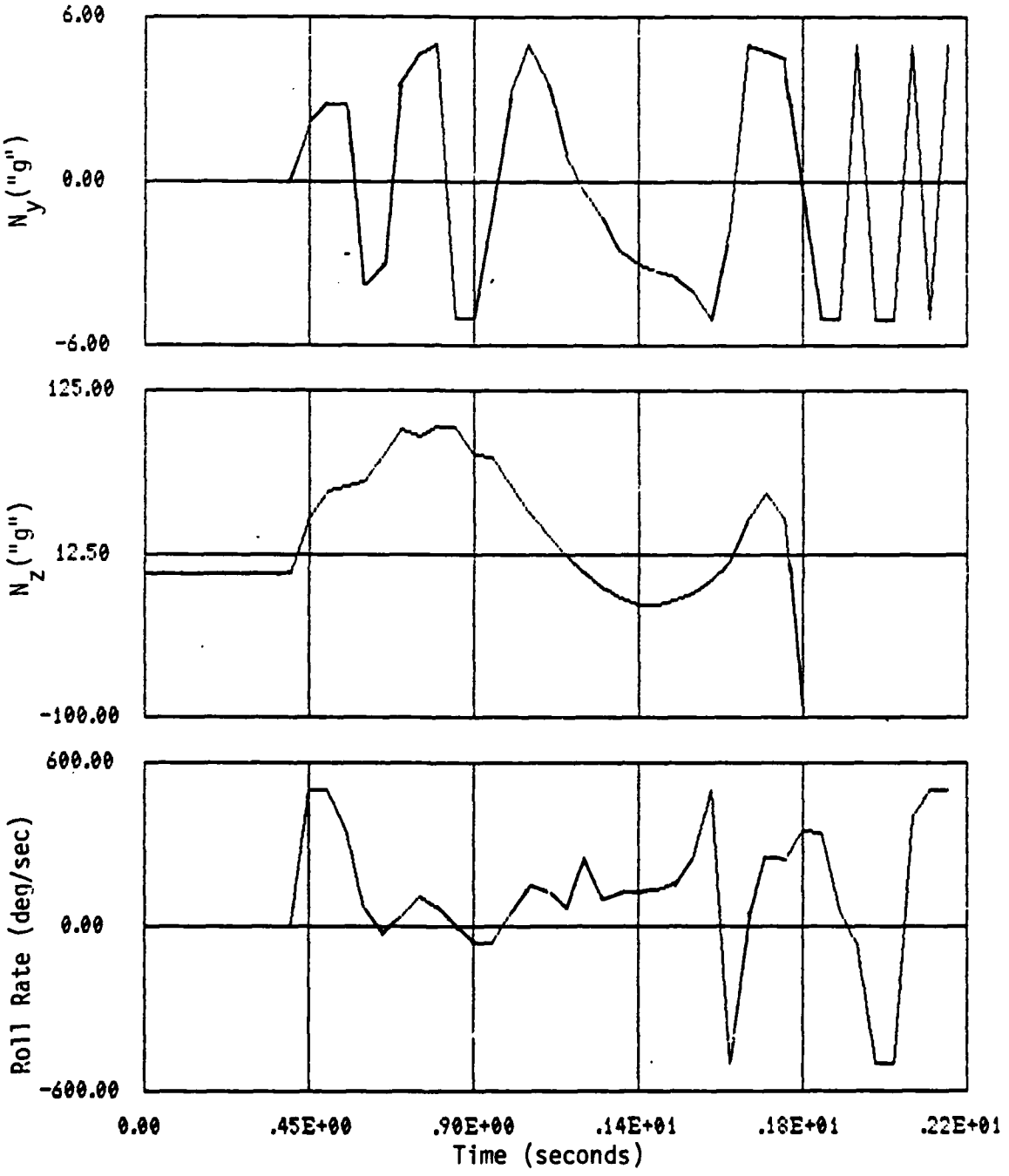


Figure 5.5B G4 MISSED INTERCEPT--ACCELERATION

Time-to-go

The incorporation of the modified Tgo algorithm significantly improved the estimate of this parameter. Since the actual Time-to-go is a function of the trajectory, and the trajectory is a strong function of Tgo, it is not possible to directly compare the algorithms. Therefore, a comparison of the actual Time-to-go and the estimate of Tgo can be made for trajectories with the same initial conditions.

After the simulated intercept, the time of the closest approach (final time) was calculated. Simulated time was then subtracted from the final time to determine the actual Time-to-go. This value is then compared to the estimate of Tgo generated during the simulation. Representative results are presented in Figures 5.5 to 5.8. In these figures, the actual and estimated Time-to-go is plotted versus the time from launch. Figures 5.5 and 5.6 came from a run with an azimuth OBA of 40° and aspect of 180° and an initial range of 7000 feet. Figures 5.7 and 5.8 are from the same geometry with an initial range of 5000 feet.

The first set of Tgo figures represents a relatively easy trajectory where the missile actually reverses its initial direction of acceleration during the intercept. Since Riggs' algorithm overestimates the closure velocity, it consistently underestimates the Time-to-go. The modified algorithm assumes that an intercept has been established; therefore, it will under or overestimate the Time-to-go depending on whether the missile is leading or lagging the target.

The Tgo plots for the 5000 foot initial conditions illustrate the performance for a more difficult trajectory. Here, the average missile acceleration is higher and it consistently lags the target. Since a true intercept condition is not established until late in the intercept,

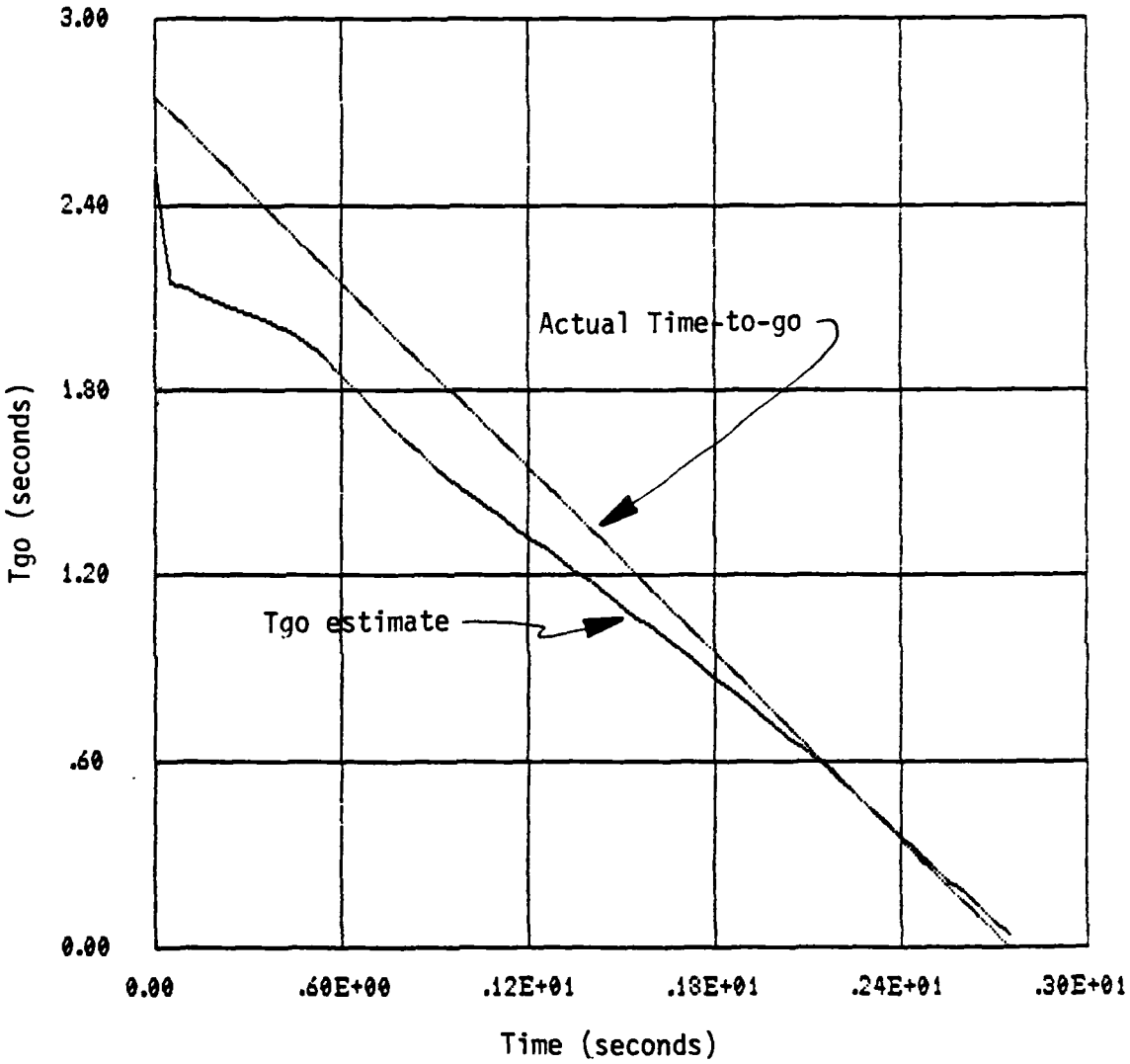


Figure 5.6 TGO PERFORMANCE--RIGGS ALGORITHM (7000 FEET)

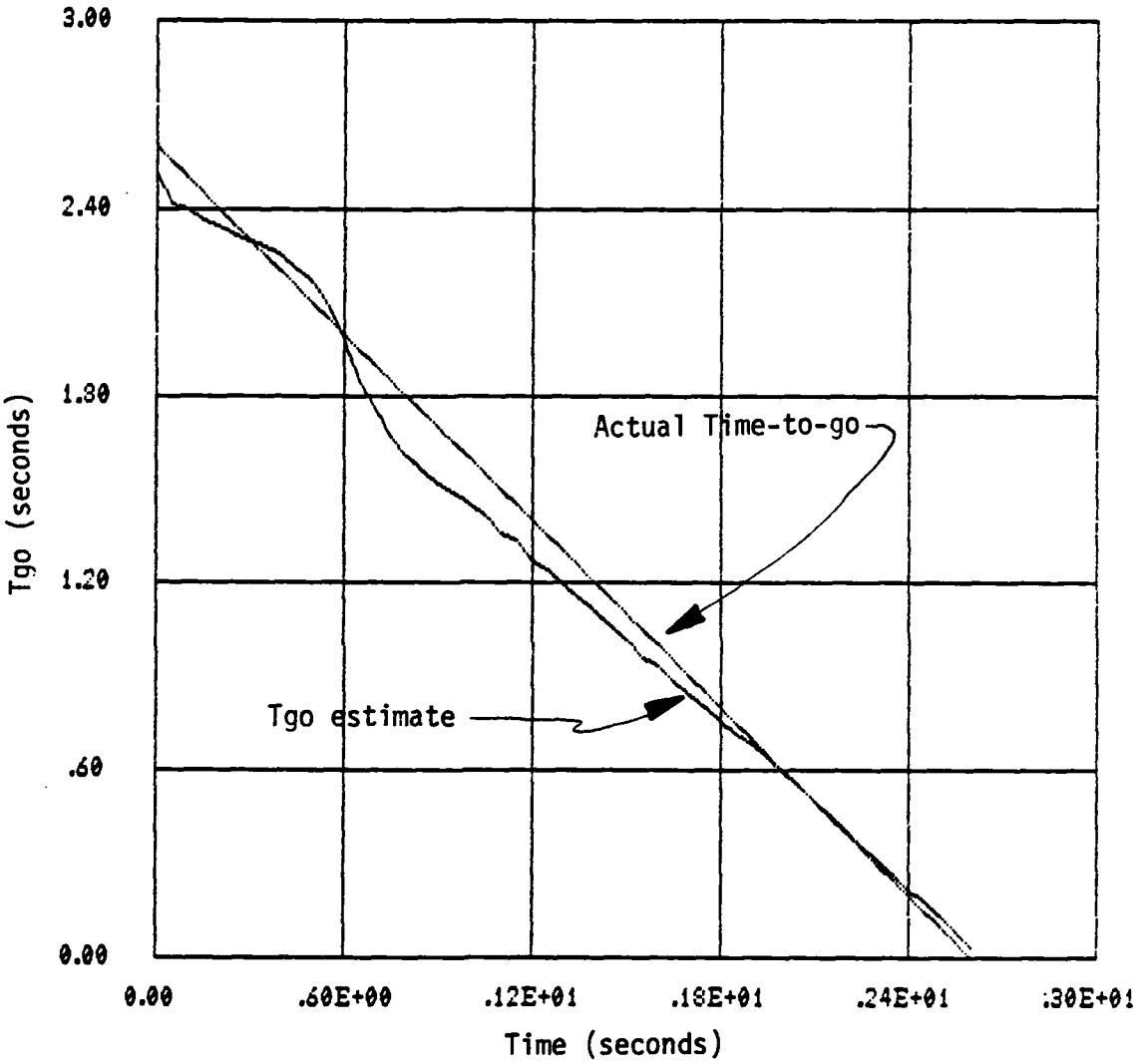


Figure 5.7 TGO PERFORMANCE--MODIFIED ALGORITHM (7000 FEET)

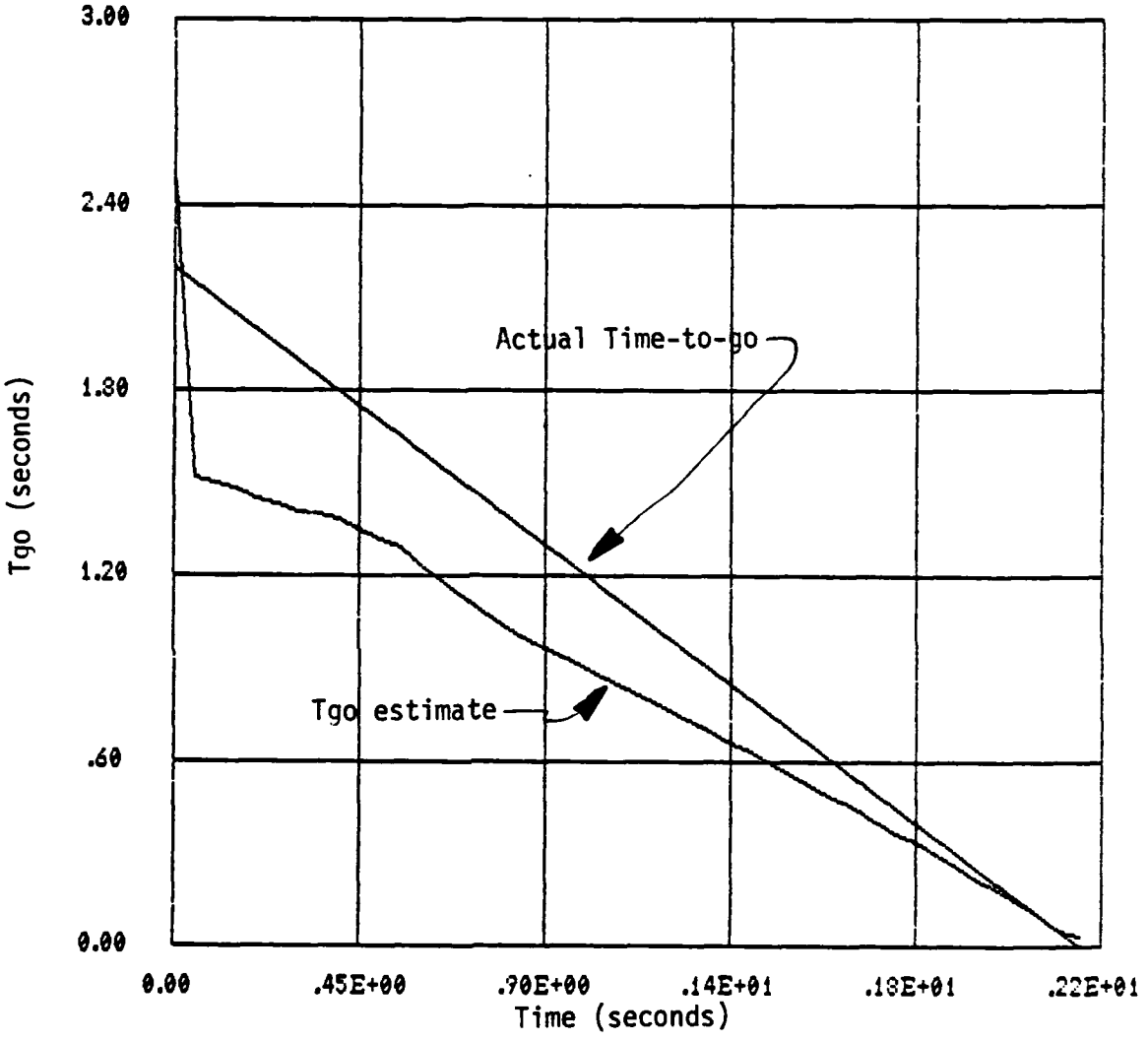


Figure 5.8 TGO PERFORMANCE--RIGGS ALGORITHM (5000 FEET)

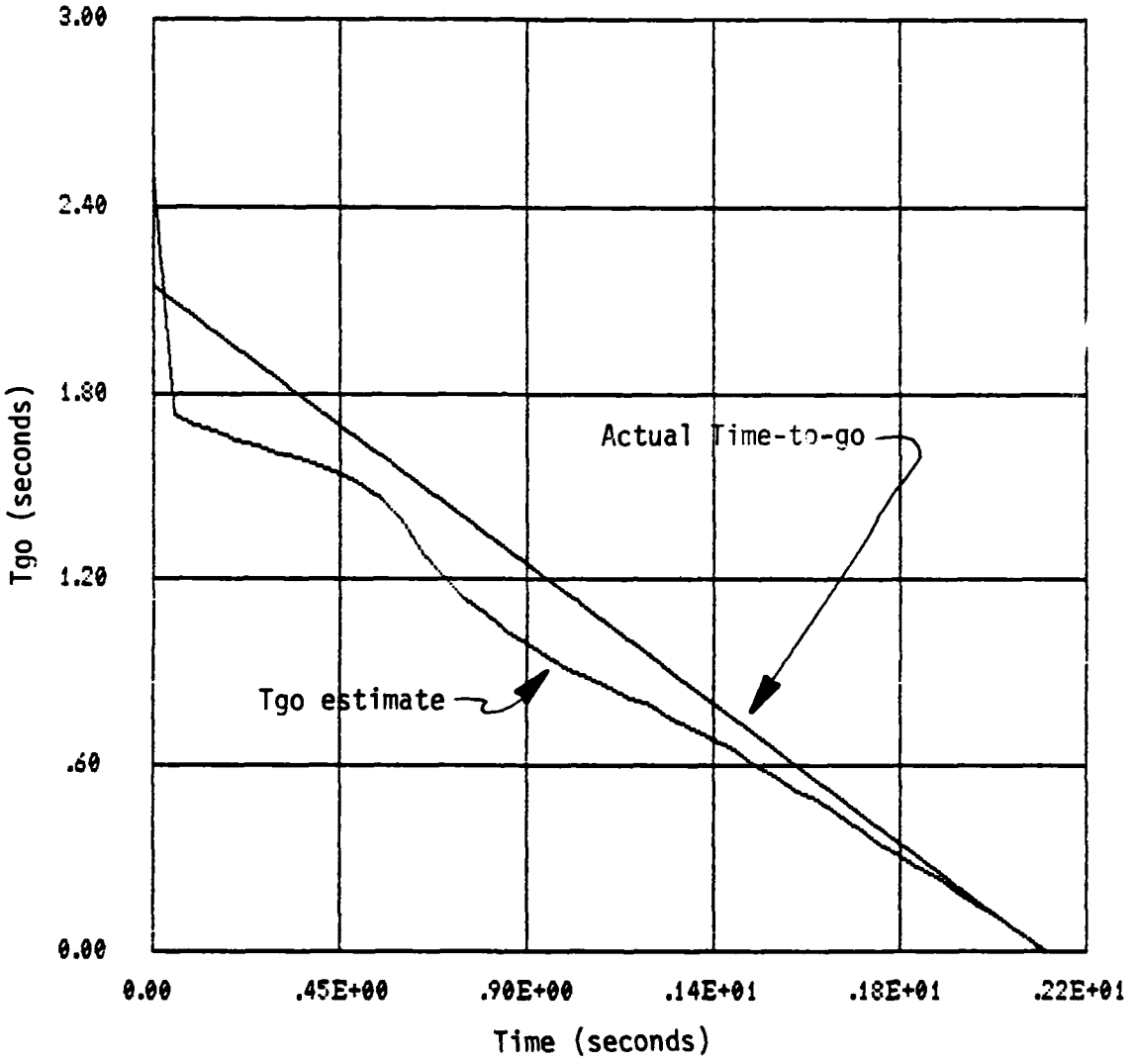


Figure 5.9 TGO PERFORMANCE--MODIFIED ALGORITHM (5000 FEET)

the absolute performance of the modified Tgo algorithm is not as good as the easier trajectory. Two points should be noted however. Just as in Riggs' algorithm, the Time-to-go is underestimated during the higher "g" trajectory when the missile is lagging the target. This error leads to higher accelerations, and if possible, an intercept. Consequently, the estimate results in a stable trajectory and the error will decay in time. Also, the absolute estimate error is always smaller than Riggs' algorithm. Table 5.3 presents the average Tgo errors for the four trajectories presented in the figures.

Table 5.3 Average Tgo Estimate Errors

Initial Range	Algorithm	Average Error
7000	Riggs'	.21
	Modified	.07
5000	Riggs'	.26
	Modified	.16

Nonlinear Control Laws

The final item to be covered is the performance of the nonlinear control law derived in Chapter 4. The control law, along with G1, (and an improved pro-nav guidance law) were incorporated in the simulation outlined in Appendix C. The simulation was verified and compared remarkably well with the larger, more complex simulation discussed previously. (The complex simulation predicted a 75 foot miss at 3500 feet versus a 193 foot miss; other results were closer.)

The trajectory presented here has the same initial geometry as the previous engagements--40 degree OBA with a 180 degree aspect angle. The initial range varies from 3400 feet to 7000 feet.

The performance of the nonlinear control law was significantly better than G1. First, the nonlinear, or Bank-to-turn (BTT), law was able to achieve a hit (miss distance less than 20 feet) inside the range of G1. Secondly, in every case where G1 was able to score a hit, the miss distance of the BTT control law was at least 25% of G1. Also, throughout the trials, the BTT accuracy was consistent while the performance of the linear guidance laws was quite erratic. And third, the time to intercept for the BTT law was less than the linear control law.

Four launch ranges are presented: 3400, 3500, 5000, and 7000 feet. Table 5.4 summarizes the results of these launches.

Table 5.4 Control Law Performance

Initial Range	Control	Miss Distance (feet)	Time (sec)
7000	G1	19	2.75
	BTT	4	2.61
5000	G1	37	2.20
	BTT	4	2.15
3500	G1	193	1.75
	BTT	5	1.65
3400	BTT	52	1.70

The details of any number of trajectories could be presented, but the most instructive are the runs from 3500 feet. Figures 5.10 and 5.10B contain the trajectory and acceleration commands for the BTT simulation, and Figures 5.11A to 5.11B have the results of G1. Since both sets of N_z commands are saturated for most of the trajectory and

BTT only deviates 22 feet in altitude, it is obvious that a hit could not be achieved from a shorter initial range unless the roll rate is increased to allow a quicker roll and turn. The reason for the improved performance is also clear. Neglecting the constraints in the guidance law results in a rolling attack as the acceleration vector is rotated by unequal limits. Consequently, it is more difficult to control the roll angle which is the real control over the direction of the acceleration vector, and roll rate commands stay large during the run. Since the BTT control law generates a ROLL command consistent with the N_z command, the ROLL angle is properly regulated, and roll rates decrease to 5 to 7 degrees per second during the terminal phase of the intercept.

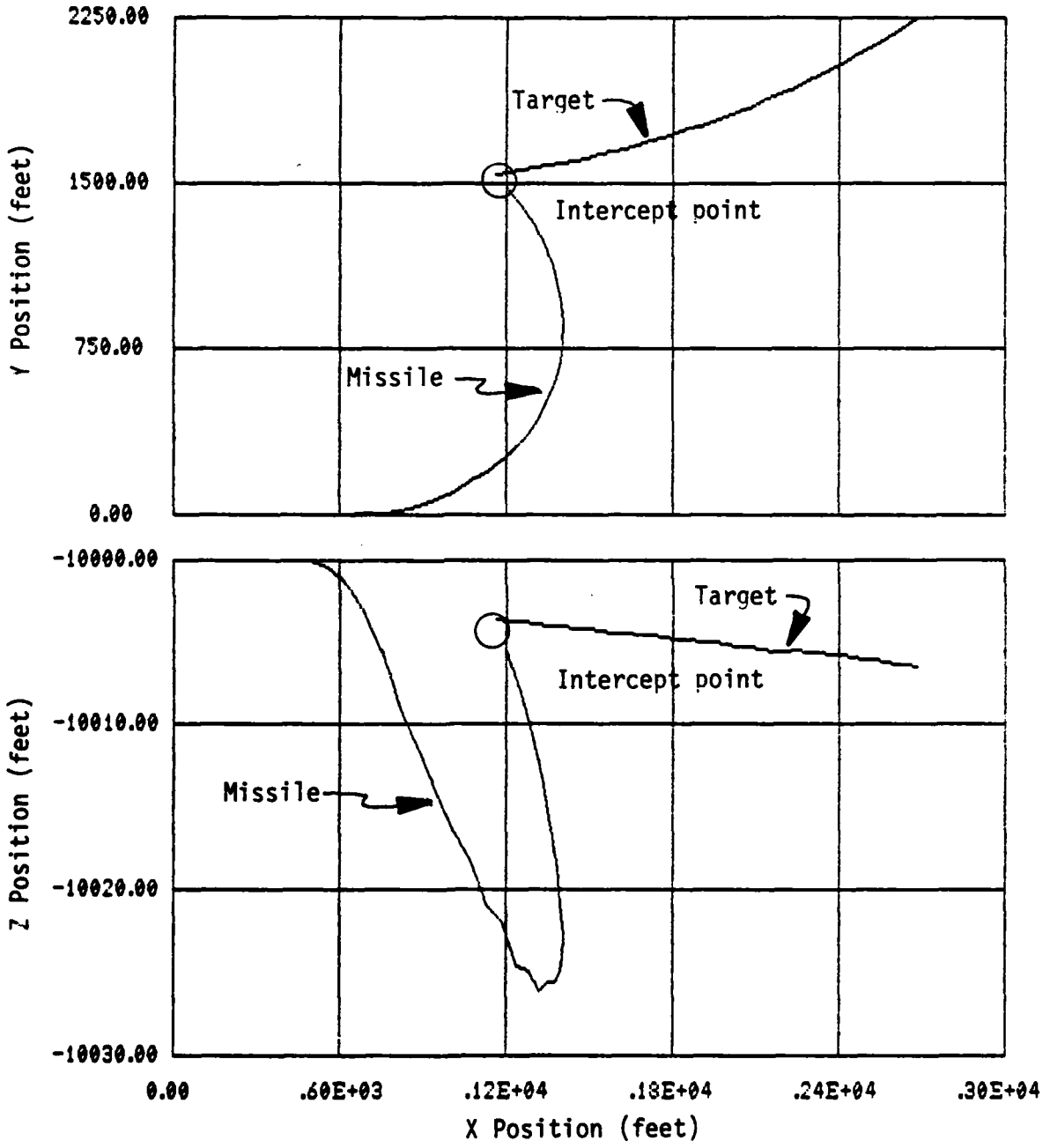


Figure 5.10A BTT TRAJECTORY

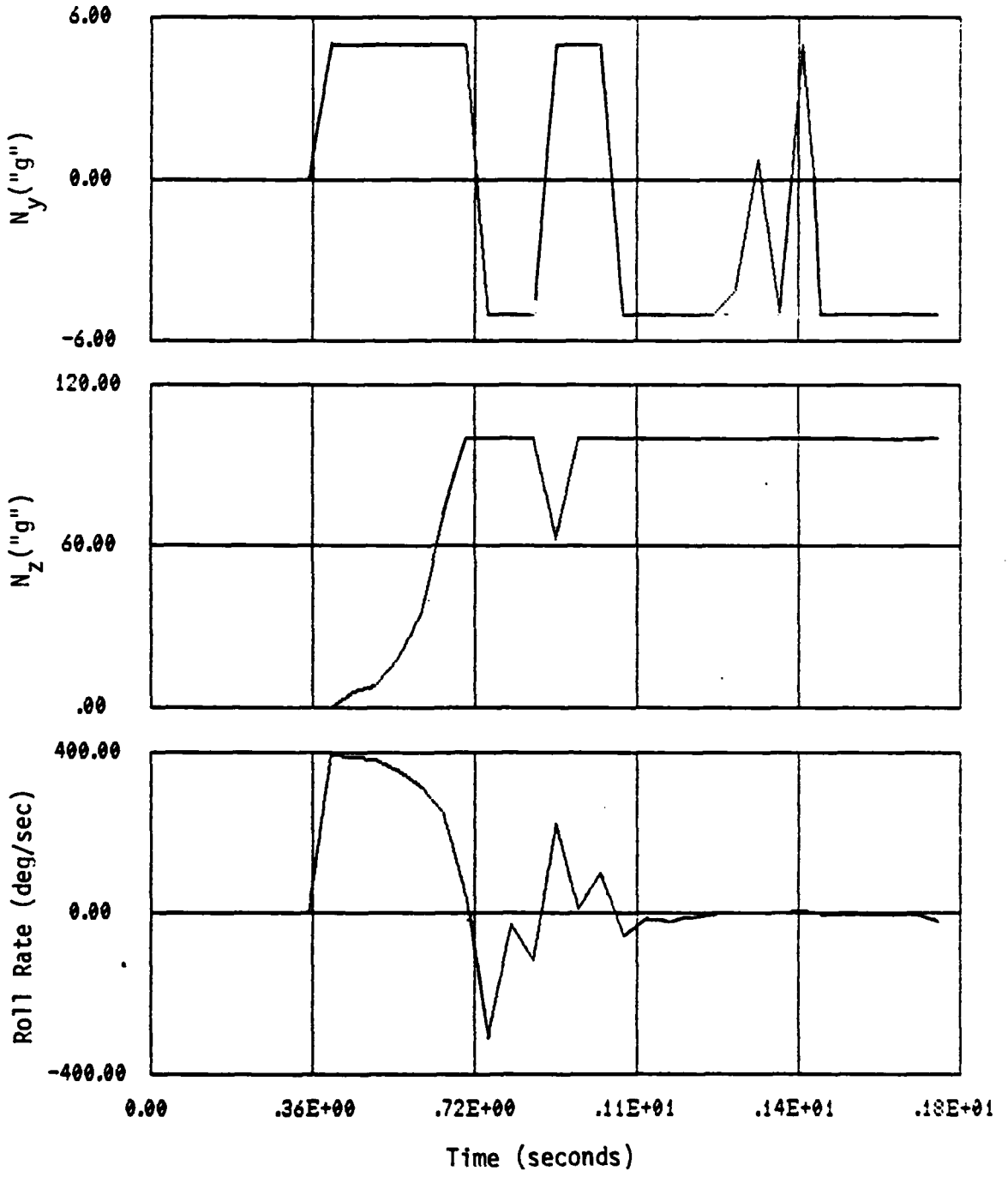


Figure 5.10B BTT ACCELERATION COMMANDS

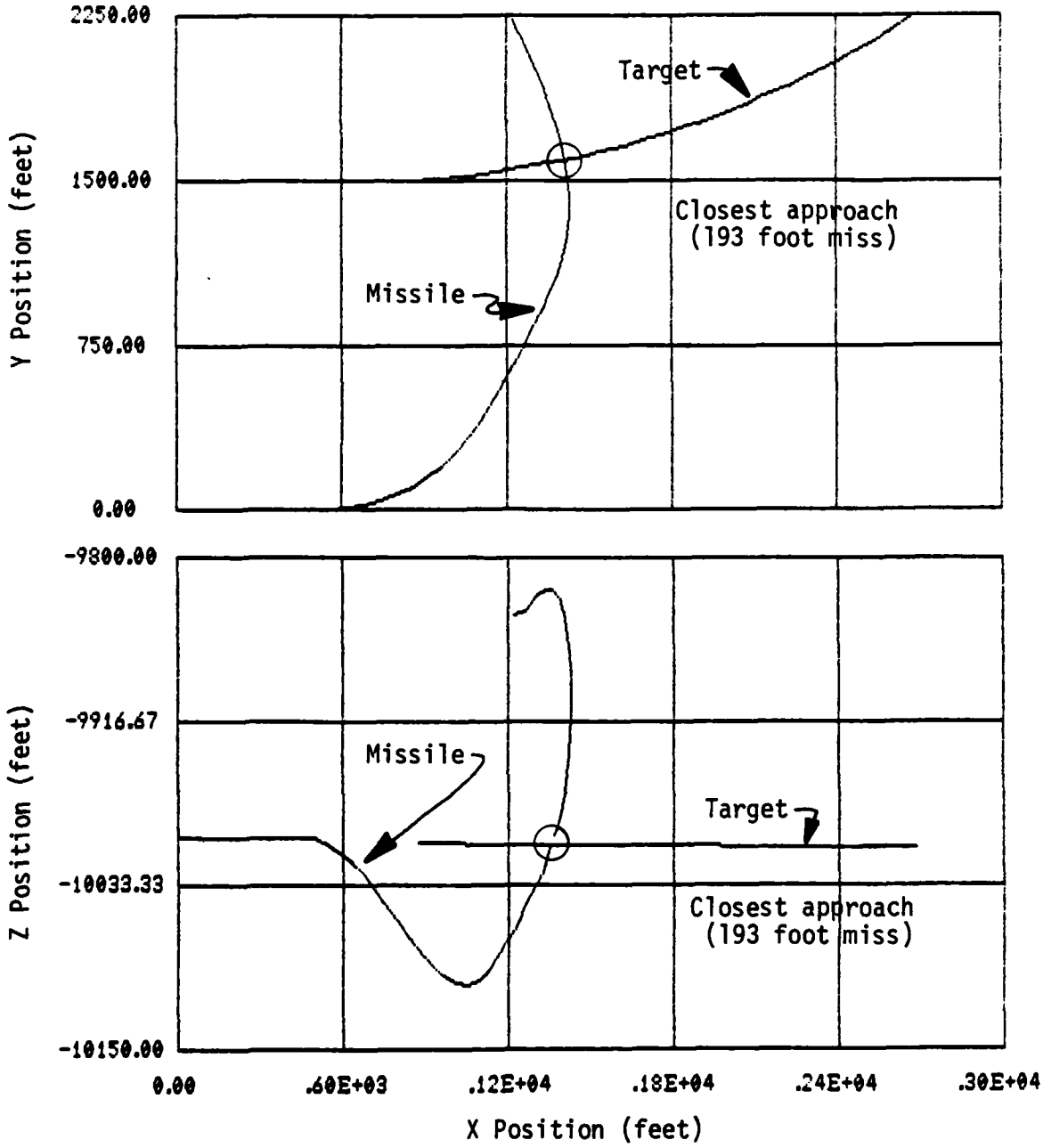


Figure 5.11A G1 TRAJECTORY

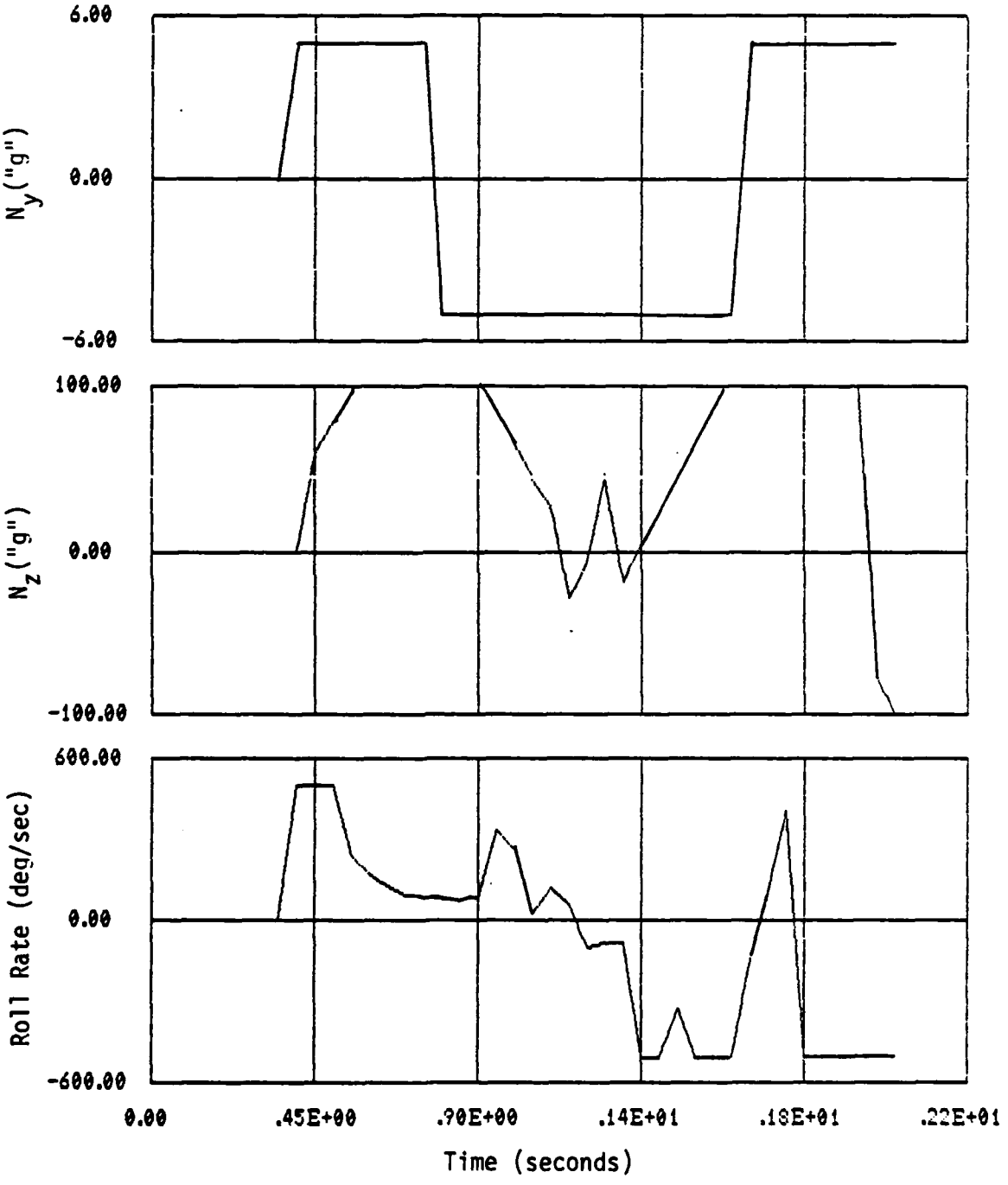


Figure 5.11B G1 ACCELERATION COMMANDS

CHAPTER VI
CONCLUSIONS

This research has identified and corrected a major shortcoming in the formulation and implementation of advanced guidance laws for a Bank-to-turn missile. The Bank-to-turn (BTT) guidance law was demonstrated on a six degree of freedom simulation developed to support the analysis of advanced guidance and control laws.

The most significant limitation of the advanced guidance techniques was the neglect of inequality constraints on the control variables. These effects were evaluated using a one-dimensional model that highlighted the effects of the constraints and identified a method to incorporate them in the formulation of the guidance law. The most significant conclusion is that while control saturation is optimal for an initial constraint, final constraints pose particular problems and must be anticipated by adjusting intermediate boundary conditions for the unsaturated portion of the trajectory.

A careful analysis of the trajectories that result from the linear guidance laws indicated that the launch envelopes determined by past research were not completely correct. In fact, an evading target could use the limitations imposed by the missile acceleration control constraints and generate a miss even in the heart of the missile launch envelope.

Rigg's algorithm, used in the computation of the Time-to-go, was improved by accounting for the angular differences between the velocity vector of the missile and the direction of the target.

Lastly, the incorporation of the first order terminal controller mentioned above as an "attitude" controller resulted in a nonlinear Bank-to-turn guidance law that exhibited exceptional performance. The guidance law was able to limit the miss distance to four or five feet inside the launch envelope, shorten the time of flight when compared to linear guidance, and decrease the inner launch boundary to a point where a near constant maximum control still resulted in a hit.

APPENDIX A
TASC LINEAR GUIDANCE LAWS

The linear guidance laws G1 and G2 developed by Fiske [8] are reproduced here to provide an indication of the type of control laws under discussion. Guidance laws G3 and G4 are an extension of G1 and G2 that add first and second order missile dynamics. As a result, they contain numerous additional terms and do not provide additional insight into the form of the guidance laws. Consequently, they are not repeated, and the reader is referred to [8]. Although Riggs and Verges [11] continued with the work of [8] and studied guidance laws GL1-GL8, these are not the same set of guidance laws but are G1 and G2 with different Tgo computation.

Since the attitude of the missile was not considered, the guidance laws are the same for each direction. Only the X direction is given, the remainder are similar.

Guidance Law--G1

$$N_X = \left\{ \frac{3(Tgo)}{3\gamma + (Tgo)^3} \right\} X + \left\{ \frac{3(Tgo)^2}{3\gamma + (Tgo)^3} \right\} V_X$$

Guidance Law--G2

$$N_x = \left\{ \frac{3(Tgo)}{3\gamma + (Tgo)^3} \right\} X + \left\{ \frac{3(Tgo)^2}{3\gamma + (Tgo)^3} \right\} V_x$$

$$+ \left\{ \frac{3(Tgo)\ e^{-\lambda_T(Tgo)} + \lambda_T(Tgo) - 1}{3\gamma + (Tgo)^3\ \lambda_T^2} \right\} A_T x$$

APPENDIX B
COORDINATE TRANSFORMATIONS

This appendix outlines the coordinate transformations used for the control laws and the simulation and is included to clarify the relationships of the various references. Unfortunately, there is no standard set of Euler angles. Confusion arises from the use of left-handed systems as well as the order, direction, axis, and reference of the rotations. The following set of Euler angles is not the "conventional" set as defined by Goldstein [24] but follows the convention outlined in [25,26,27].

Definitions

Missile Euler Angles (Figure 5.1)

Those rotations, in the order of YAW, PITCH, and ROLL, are required to re-orient the missile from the inertial reference frame to the current body frame orientation. The inertial reference is defined as the orientation of the body axis at time zero.

The body axis system is fixed to the center of gravity of the missile with the positive x axis pointing forward along the fuselage reference line. The positive z axis points downward toward the bottom of the vehicle, with the y axis completing the right handed system (out the right side).

YAW ϕ: The angle between the projection of the x body axis onto the inertial XY plane and the X inertial axis.

PITCH θ: The angle measured in the vertical plane between the x body axis and the inertial XY plane.

ROLL ϕ: (Bank) Angle measured in the yz plane of the body system between the y body axis and the YX inertial plane.

Seeker Angles (Figure B.2)

The orientation of the seeker line of sight (LOS) and missile body axis are defined by seeker gimbal azimuth and elevation angles ψ_g and θ_g, respectively.

GMB PSI ψ_g: The angle between the projection of the LOS onto the missile xy plane and the X body axis.

GMB THT θ_g: The angle between the LOS and the missile xy plane.

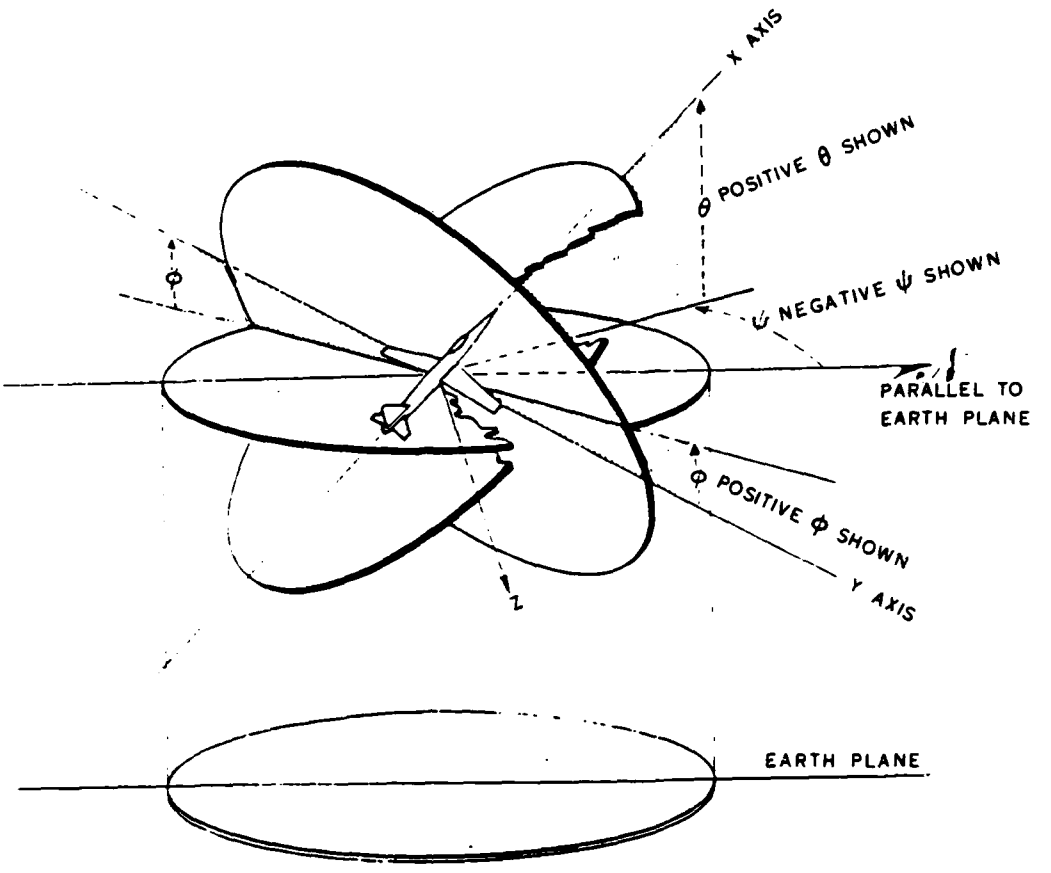


Figure B.1 MISSILE REFERENCE SYSTEM

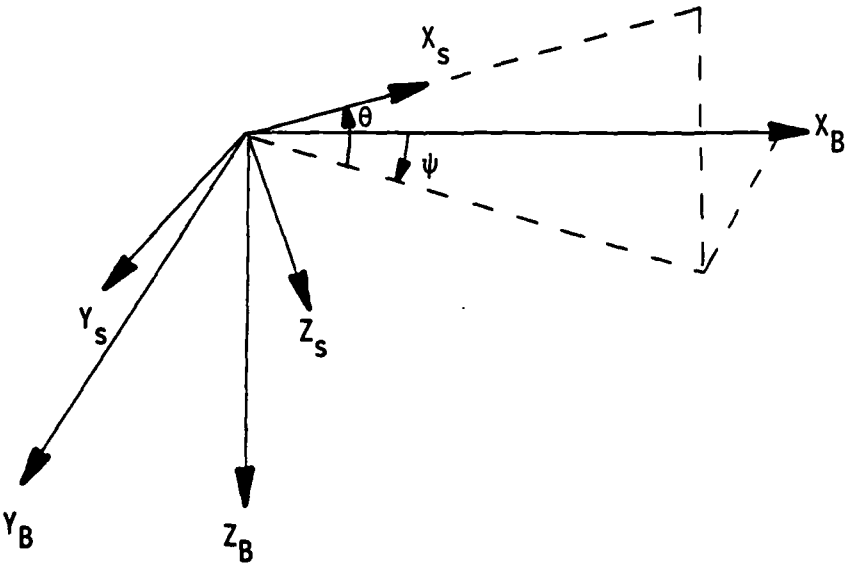


Figure B.2 SEEKER REFERENCE SYSTEM

Transformations

Missile Inertial to Body Axis

YAW: YAW is a clockwise rotation about the inertial Z axis while looking in the positive Z direction.

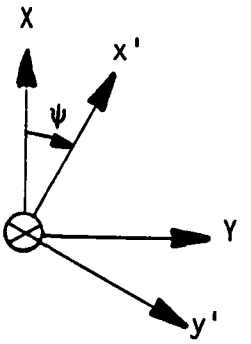


Figure B.3 YAW ANGLE ROTATION

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

PITCH: PITCH is a clockwise rotation about the current y' body axis while looking in the positive y' direction.

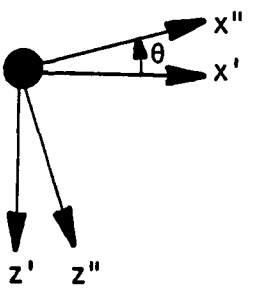


Figure B.4 PITCH ANGLE ROTATION

$$\begin{pmatrix} x'' \\ y'' \\ z'' \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}$$

ROLL: ROLL is a clockwise rotation about the current x" axis while looking in the positive x" direction.

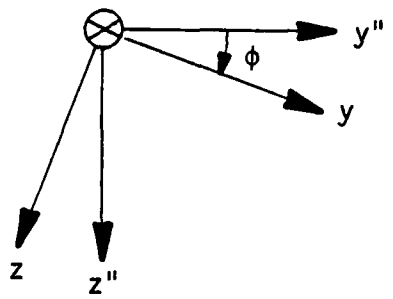


Figure B.5 ROLL ANGLE ROTATION

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} x'' \\ y'' \\ z'' \end{pmatrix}$$

Combining the three rotations in the proper order generates the general transformation from inertial to body coordinates:

$$D = \begin{matrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{matrix}$$

Since the transformation is orthogonal, the transformation from body to inertial is the transpose of D.

Missile Body to Seeker LOS

Seeker Azimuth: The seeker azimuth angle, gimbal psi (ψ_g), is a rotation about the z body axis while looking in the positive z direction.

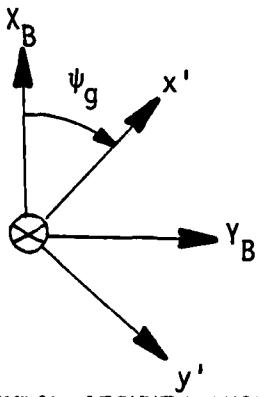


Figure B.6 GIMBAL AZIMUTH ANGLE ROTATION

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos \psi_g & \sin \psi_g & 0 \\ -\sin \psi_g & \cos \psi_g & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Seeker PITCH: The seeker PITCH, gimbal theta (θ_g), is a rotation about the y' seeker axis while looking in the positive y' direction.

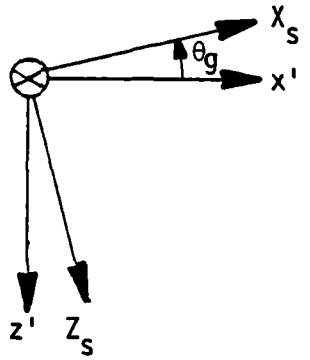


Figure B.7 GIMBAL PITCH ANGLE ROTATION

$$\begin{pmatrix} Xs \\ Ys \\ Zs \end{pmatrix} = \begin{pmatrix} \cos \theta_g & 0 & -\sin \theta_g \\ 0 & 1 & 0 \\ \sin \theta_g & 0 & \cos \theta_g \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}$$

Consequently, the final transformation from the body to seeker reference is:

$$SEEK = \begin{pmatrix} \cos\theta_g \cos\psi_g & \cos\theta_g \sin\psi_g & -\sin\theta_g \\ -\sin\psi_g & \cos\psi_g & 0 \\ \sin\theta_g \cos\psi_g & \sin\theta_g \sin\psi_g & \cos\theta_g \end{pmatrix}$$

APPENDIX C
SMALL SIMULATION

Appendix C presents the approximate equations of motion and code for the first order missile simulation. The appendix is divided into three sections: the equations of motion, the seeker model, and the simulation.

Missile Equations of Motion

The first section contains the derivation of the approximate equations of motion used in the simulation to transform control inputs to inertial Euler angle rates.

Using the Euler angle definitions of Appendix B, a transformation can be derived that accounts for the effect of the control inputs, linear accelerations, and roll rate on the inertial attitude of the missile [26,29].

Two sets of relationships are required. The first is the transformation between the body axis rates: p, q, and r, and inertial Euler angle rates $\dot{\phi}$ $\dot{\theta}$ $\dot{\phi}$. The second relationship is rhe response of the missile to input commands. The following derivation follows [26].

Definitions

The linear velocity of the missile, expressed in terms of body axis coordinates is: $V = u\hat{i} + v\hat{j} + w\hat{k}$

where u = forward velocity
v = side velocity
w = vertical velocity

The angular velocity of the missile also expressed in the body reference system is:

$$w = p\hat{i} + q\hat{j} + r\hat{k}$$

where p = ROLL rate

q = PITCH rate

r = YAW rate

Angular Velocity Transformation

Because the Euler angles are defined in YAW, PITCH, ROLL sequence, this sequence must be maintained in determining the angular velocity transformation.

Resolve components of YAW rate $\dot{\psi}$

In the case of straight and level flight, the inertial axis system remains aligned with the z body axis for all ψ, therefore,

$$r = \dot{\psi}.$$

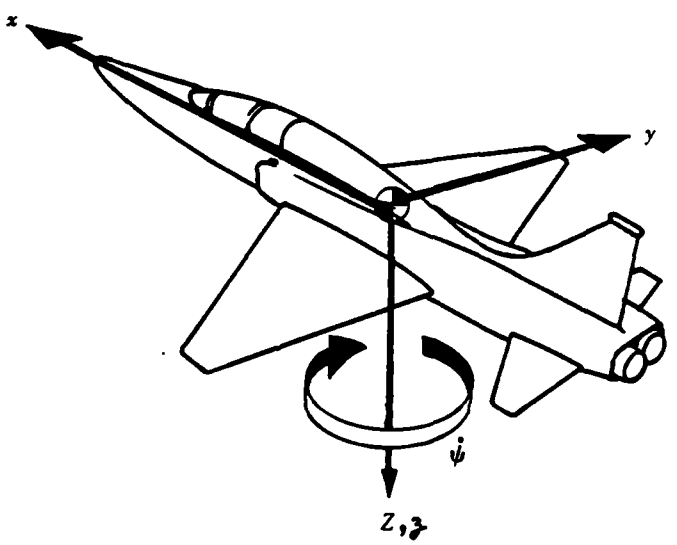


Figure C.1 YAW RATE COMPONENTS--LEVEL

With a positive PITCH attitude, the missile $\dot{\psi}$ has components on both the x and z body axis while the pitch rate is not affected.

$$p = -\dot{\psi} \sin \theta$$

$$r = \dot{\psi} \cos \theta$$

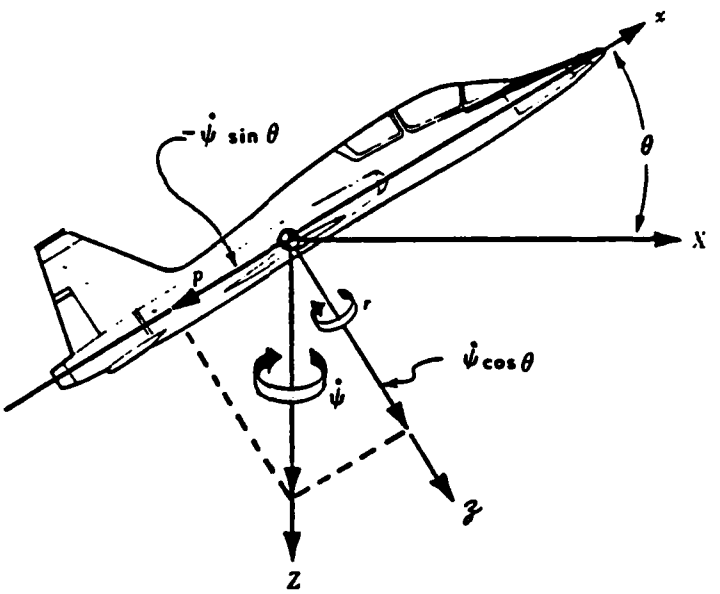


Figure C.2 YAW RATE COMPONENTS--PITCH UP

A ROLL angle will not change the effect of $\dot{\psi}$ on p, and the PITCH and ROLL rate components are:

$$q = \dot{\psi} \cos \theta \sin \phi$$

$$r = \dot{\psi} \cos \theta \cos \phi$$

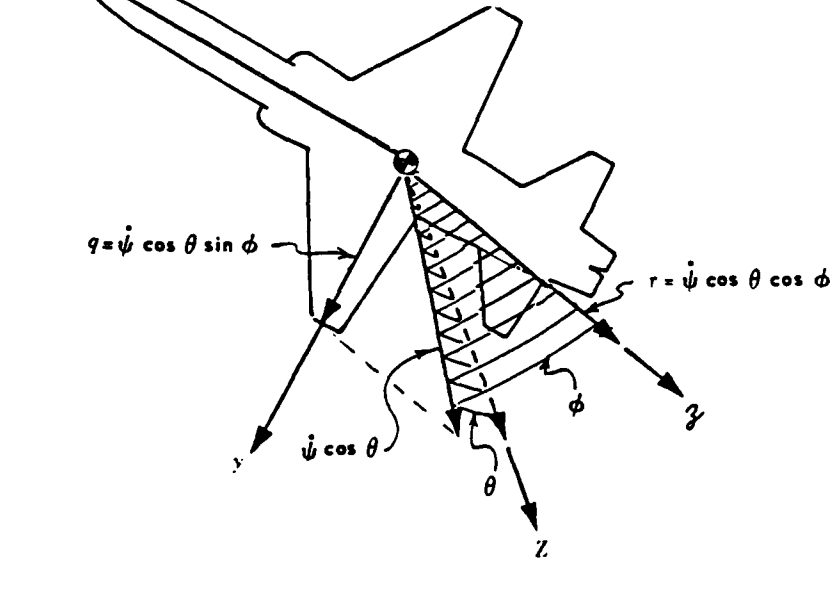


Figure C.3 YAW RATE COMPONENTS--ROLL

Resolve components of PITCH rate $\dot{\theta}$

With the missile in a positive PITCH attitude, it can be seen immediately that the YAW angle will not have any effect on the resolution of the inertial PITCH rate. Also, when pitched up, the y body axis remains in the X-Y inertial plane. Consequently, an inertial PITCH rate will equal the body axis rate:

$$q = \dot{\theta}$$

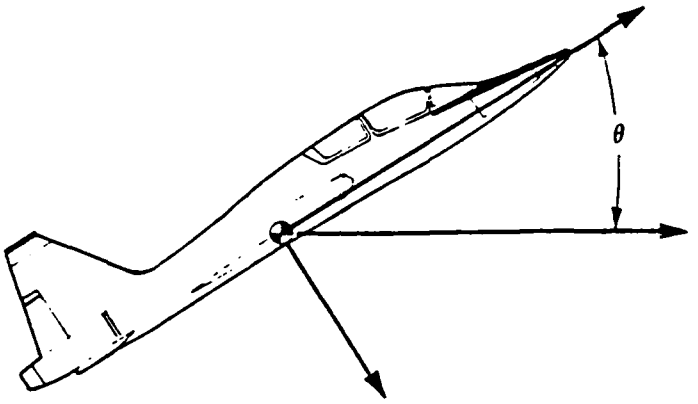


Figure C.4 PITCH RATE COMPONENTS--PITCH

Once the missile has a nonzero ROLL angle, components of $\dot{\theta}$ will show up as:

$$q = \dot{\theta} \cos \phi$$

$$r = -\dot{\theta} \sin \phi$$

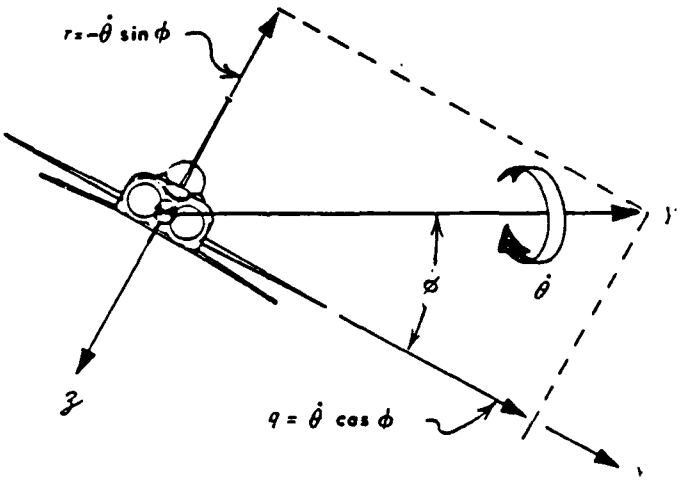


Figure C.5 PITCH RATE COMPONENTS--ROLL

Note that the body axis ROLL rate p is not affected by $\dot{\theta}$ since $\dot{\theta}$ is measured with respect to an axis orthogonal to the x body axis.

Resolve the components of ROLL rate $\dot{\phi}$

Since $\dot{\phi}$ is measured along the x body axis by definition, it will affect the value of p only.

$$p = \dot{\phi}$$

Combined Transformation

The components of the inertial angular velocities with respect to the missile body axis can be combined to give the following transformation:

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}$$

Inverse Transformation

In order to account for the affect of the control inputs on the attitude of the missile, the inverse of the above transformation is needed. The determinant of the transformation matrix, however, is equal to the cos θ. Consequently, at ±90 degrees of PITCH, the inverse of the transformation is not defined, but from the geometry of the transformations, the inverse can be seen to be:

$$\dot{\psi} = p$$

$$\dot{\theta} = q$$

$$\dot{\phi} = r$$

When the cos $\theta \neq 0$, then the inverse of the transformation can be computed as:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}$$

Missile Dynamics

In order to determine the effect of control inputs, the equations of motion of the missile must be analyzed. Since the missile has six (6) degrees of freedom (forward, sideways, and down translation as well as the PITCH, ROLL, and YAW rotation), six simultaneous equations are required. Using the linear force and moment relationships expressed in an accelerating reference frame (with $(dV/dt)_I = (dV/dt)_B + \omega \times V$), the following equations can be derived [25,26]:

$$F_x = m(\dot{u} + qw - rv)$$
$$F_z = m(\dot{w} + pv - qu)$$
$$F_y = m(\dot{v} + ru - pw)$$
$$G_y = \dot{q}\, Iy - pr\,(Iz - Ix) + (p^2 - r^2)\, Ixz$$
$$G_x = \dot{p}\, Ix + qr\,(Iz - Iy) - (\dot{r} + pq)\, Ixz$$
$$G_z = \dot{r}\, Iz + pq\,(Iy - Ix) - (qr - \dot{p})\, Ixz$$

Here, the left hand side represents applied forces (F_i) and moments (G_i) while the right hand side provides the missile response as a function of moments of inertia and body referenced velocities.

Equating the applied accelerations to the control inputs, and incorporating the inverse transformation ($\theta \neq 90°$):

$$\begin{pmatrix} Nx \\ Nz \\ Ny \end{pmatrix} = \begin{pmatrix} \dot{u} \\ \dot{w} \\ \dot{v} \end{pmatrix} + \begin{pmatrix} 0 & w & -v \\ v & -u & 0 \\ -w & 0 & u \end{pmatrix} \begin{pmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}$$

This set of equations, however, cannot be used to determine the attitude dynamics resulting from control inputs since the inverse of:

$$\begin{pmatrix} 0 & w\cos\phi + v\sin\phi & w\sin\phi\cos\theta - v\cos\phi\cos\theta \\ v & -u\cos\phi & -v\sin\theta - u\sin\phi\cos\theta \\ -w & -u\sin\phi & w\sin\theta - u\cos\phi\cos\theta \end{pmatrix}$$

does not exist.

Using the applied accelerations:

$$F_x/m = \dot{u} + qw - rv$$

$$F_z/m = \dot{w} + pv - qu$$

$$F_y/m = \dot{v} + ru - pw$$

and restricting the missile velocities to the x (forward) direction only, then

$$\dot{v} = v = \dot{w} = w = 0$$

and $N_x = \dot{u}$ $N_z = -qu$ $N_y = ru$

These assumptions remove higher order dynamics about the center of gravity of the missile (Dutch roll, Short Period, Phugoid Oscillations, and Directional/Spiral Divergence) but still allow dynamics associated with the center of gravity.

Using these approximations and the inverse angular velocity transformation, we have the following relationships:

$$\dot{\phi} = p - \sin\phi \tan\theta\,(Nz/u) + \cos\phi \tan\theta\,(Ny/u)$$

$$\dot{\theta} = -\cos\phi\,(Nz/u) - \sin\phi\,(Ny/u)$$

$$\dot{\psi} = -\sin\phi/\cos\theta\,(Nz/u) + \cos\phi/\cos\theta\,(Ny/u)$$

Seeker Model

While the above relationships provide the necessary equations to model the missile motions, the missile actually uses a seeker to determine the relative position of the target. Consequently, a seeker model is required. Rather than model the internal motions of the seeker gimbals, seeker error sources, and the stabilization loops [7], a simplified approach is taken where the LOS angles are propagated via:

GMB Theta = DLOSQ - ω_y

GMB Psi = DLOSR - ω_z

where DLOSQ = PITCH line of sight rate

DLOSR = azimuth line of sight rate

ω_y = Missile motion affecting θ_g

ω_z = Missile motion affecting ϕ_g

The LOS errors are determined by calculating the difference in the predicted target location (using Gimbal angles), and the actual target location (using simulation results). Assuming a type 1 system, the feedback transfer function from θ_g (or ϕ_g) to tracking angle error ε (assuming perfect stabilization and no abberations) is

$$\frac{\varepsilon}{\theta_g} = \frac{\tau_1 s}{1 + \tau_1 s}$$

Then, at low frequencies, the LOS rate is proportional to the tracking error [7]. Therefore, the gimbal errors are used with a proportionality constant to determine the LOS rate. Gimbal motion due to missile angular velocities, ω_y and ω_z, are simply body rates transformed to the seeker coordinate system.

Simulation

The simulation consists of a 15 state model that is propagated using an Adams Predictor-Corrector [19,28] with a Runge-Kutta start up. The equations of motion are contained in the subroutine MODEL while other routines are used for initialization and output. The states are relative cartesian positions and velocities, target ROLL angles, missile Euler angles, and missile gimbal angles. Inputs are linear accelerations N_x, N_y, N_z, and missile ROLL rate.

The simulation is divided into three (3) parts so that it can be run on a microcomputer in 40K COM files. The code, however, is ANSI standard FORTRAN and with two changes for file handling, was run on a VAX 1170. The first part initialized the variables and simulated the trajectory. Data were stored in two data files W1.DAT and PLOT.DAT. These files were read by the second and third programs for data reporting, postflight miss distance, Tgo analysis and plotting. Since most of the subroutines are straightforward input-output, they are not included. The code used to initialize the routines and the subroutine MODEL is included for illustrative purposes.

As mentioned earlier, the simulation had 15 states: the usual nine (9) for cartesian positions, velocities, and accelerations, two (2) to propagate gimbal angle dynamics, one (1) (unused) for target ROLL angle,

and the remaining three (3) for the missile Euler angles. Linear states are propagated without the effects of inertia or drag. But if the angular states were also propagated by assuming an instantaneous and perfect response without damping, excessive rates would develop. Therefore, it became obvious that the bandwidth of the angular velocities had to be limited in order to produce reasonable results. Since the effects of the linear accelerations on the angular rates no included a delay, the direction of the missile indicated by missile velocities and Euler angles was no longer consistent. Since each was propagated independently, by the end of the run, a three to five degree discrepency would develop between them. In effect, there were too many states. Consequently, at each update, the Euler angles are corrected to agree with the angles generated by the linear velocity terms.

```
      SUBROUTINE MODEL(TIME,YONE,YPRIME,HIT)
      IMPLICIT REAL*8 (A-I,K-Z)
      LOGICAL HIT
      REAL PLOTX1,PLOTX2,PLOTY1,PLOTY2,PLOTZ1,PLOTZ2
      REAL MINX,MINY,MAXX,MAXY
      INTEGER I,N,JSTATE,JINPUT,JCNTRL,JTIME,JLIMIT,JROLL,JOB
      INTEGER INCHX,INCHY,NPLOT,NPTS
      DIMENSION YONE(15),YPRIME(15)
      COMMON /PLOT/PLOTX1(25),PLOTX2(25),PLOTY1(25),PLOTY2(25),
     1PLOTZ1(25),PLOTZ2(25),NPLOT,NPTS
      COMMON /TRGT/TRG XI,TRG YI,TRG ZI,
     1TRG VXI,TRG VYI,TRG VZI,
     2TRG AXI,TRG AYI,TRG AZI
      COMMON /SET/ALTID0,GMB TH0,GMB PS0,RNG0,TRG V0,
     1ASP TH0,ASP PS0,MSL V0,THET0,PHI0,PSI0,TH0,CSPSI0,SNPSI0,
     2SNTH0,MISS DS,STEP,DLAY SW,TRG XI0,TRG YI0,TRG ZI0,
     3TXO,TYO,TZO,OLD T,TRG PS0,TRG TH0,TRG PHI,JMAN
      COMMON /MSL/Y(15),U(4),TGO HT,RNG I,VC
      COMMON /OUT/W1(25,25)
      COMMON /SEEK/ SEEK 11,SEEK 12,SEEK 13,SEEK 21,
     1SEEK 22,SEEK 23,SEEK 31,SEEK 32,SEEK 33,DLOSQ,DLOSR
      COMMON /OPTION/ JCNTRL,JTIME,JLIMIT,JROLL
C
C     TRIG FUNCTIONS FOR DEGREES
C
      SIN DG(X)   = DSIN(DG TO RD * X)
      COS DG(X)   = DCOS(DG TO RD * X)
C
      GRAV     = 32.1725
      DG TO RD = .0174532925
      RD TO DG = 57.29577951
      PI       = 3.141592654
      TWOPI    = 6.283185307
      E = 2.7182818
C
C     TARGET CONSTANTS
C
      NMAX    = 9.0
C
C     MISSILE/AUTOPILOT CONSTANTS
C
      W PHI    = 15.
      W THETA  = 15.
      W PSI    = 15.
      LAMDA X  = 10.
      LAMDA Y  = 10.
      LAMDA Z  = 10.
      AMAX     = 580.
      AMIN     = -38.
      DURAT    = 2.6
```

```
      DLAY SW = 0.4
      GAMMA   = 1E-4
      NZ MAX  = 100.*GRAV
      NY MAX  = 5.*GRAV
      P MAX   = 500.*DG TO RD
C
      MSL XI    =  YONE(1)
      MSL YI    =  YONE(2)
      MSL ZI    =  YONE(3)
      MSL VXI   =  YONE(4)
      MSL VYI   =  YONE(5)
      MSL VZI   =  YONE(6)
C
C     LIMIT ANGLES TO 2 * PI
C
      PHI       = YONE(8)

C
C     CONSTRAIN PSI & THETA TO VELOCITIES
C
      MSL IXY   = DSQRT((MSL VXI**2) + (MSL VYI**2))
      PSI       = DATAN2(MSL VYI,MSL VXI)
      THETA     = DATAN2(-MSL VZI,MSL IXY)

      YONE(9)   = THETA
      YONE(10)  = PSI
      D PHI     = YONE(11)
      D THETA   = YONE(12)
      D PSI     = YONE(13)
      GMBTHT    = YONE(14)
      GMBPSI    = YONE(15)
      IF(DABS(PHI).GT.TWOPI)PHI = PHI -
     1DSIGN(TWOPI,PHI)*DINT(TWOPI/PHI)
      IF(DABS(THETA).GT.TWOPI)THETA = THETA -
     1DSIGN(TWOPI,THETA)*DINT(TWOPI/THETA)
      IF(DABS(PSI).GT.TWOPI)PSI = PSI -
     1DSIGN(TWOPI,PSI)*DINT(TWOPI/PSI)
      IF(DABS(GMB THT).GT.TWOPI)GMB THT = GMB THT -
     1DSIGN(TWOPI,GMB THT)*DINT(TWOPI/GMB THT)
      IF(DABS(GMB PSI).GT.TWOPI)GMB PSI = GMB PSI -
     1DSIGN(TWOPI,GMB PSI)*DINT(TWOPI/GMB PSI)
C
C     INERTIAL TO BODY DIRECTION COSINES
C
      D11 = DCOS(THETA)*DCOS(PSI)
      D12 = DCOS(THETA)*DSIN(PSI)
      D13 = - DSIN(THETA)
C
      D21 = DSIN(PHI)*DSIN(THETA)*DCOS(PSI) -
     1      DCOS(PHI)*DSIN(PSI)
```

```
      D22 = DSIN(PHI)*DSIN(THETA)*DSIN(PSI) +
     1      DCOS(PHI)*DCOS(PSI)
      D23 = DSIN(PHI)*DCOS(THETA)
C
      D31 = DCOS(PHI)*DSIN(THETA)*DCOS(PSI) +
     1      DSIN(PHI)*DSIN(PSI)
      D32 = DCOS(PHI)*DSIN(THETA)*DSIN(PSI) -
     1      DSIN(PHI)*DCOS(PSI)
      D33 = DCOS(PHI)*DCOS(THETA)
C
C     COMPUTE BODY COORD VELOCITY (BODY TO INERTIAL)
C
      MSL VXB = DSQRT((MSL VXI**2) + (MSL VYI**2) + (MSL VZI**2))
C
C     CHECK FOR SINGULAR INVERSE
C
      IF(DABS(THETA-(PI/2.)).LE.DG TO RD) GOTO 100
C
      A11   = DSIN(PHI)*DSIN(THETA)/(DCOS(THETA)*MSL VXB)
      A12   = DCOS(PHI)*DSIN(THETA)/(DCOS(THETA)*MSL VXB)
      A21   = DCOS(PHI)*(1/MSL VXB)
      A22   = DSIN(PHI)*(1/MSL VXB)
      A31   = DSIN(PHI)/(DCOS(THETA)*MSL VXB)
      A32   = DCOS(PHI)/(DCOS(THETA)*MSL VXB)
C
  100 CONTINUE
C
C     COMPUTE PITCH RATES IN BODY AXIS
C
      PM    = D PHI -  D PSI*DSIN(THETA)
      QM    = D THETA*DCOS(PHI) + D PSI*DSIN(PHI)*DCOS(THETA)
      RM    =-D THETA*DSIN(PHI) + D PSI*DCOS(PHI)*DCOS(THETA)
C
C     **********************************************
C     ****                                      ****
C     ****      Target Equations of Motion      ****
C     ****                                      ****
C     **********************************************
C
      IF(RNG I.LE.6000.)GOTO 200
C
      TRG XI = TRG VXI * TIME + TRG XI0
      TRG YI = TRG VYI * TIME + TRG YI0
      TRG ZI = TRG VZI * TIME + TRG ZI0
C
      TXO    = TRG XI
      TYO    = TRG YI
      TZO    = TRG ZI
C
      OLD T  = TIME
```

```
      JMAN    = 0
C
      GOTO 300
C
  200 CONTINUE
C
      IF(JMAN.EQ.1) GOTO 250
C
C     INITIALIZE MANUEVER
C
C     COMPUTE TARGET ROLL ANGLE
C
C      + TRG PHI = RIGHT ROLL
C      - TRG PHI = LEFT ROLL
C
      IF(TRG PHI.GE.180.) TRG PHI = TRG PHI - 360.
      IF(TRG PHI.LE.-180.) TRG PHI = TRG PHI + 360.
C
C     DETERMINE THE RADIUS OF CURVATURE & ANGULAR VELOCITY
C
      TRG R = (TRG V0**2)/(NMAX*GRAV)
      PS DOT = RD TO DG * TRG V0/TRG R
C
C     COMPUTE DIR COSINES FOR AXIS OF ROTATION
C
C                       BODY TO INERTIAL
C
      DT11 =COS DG(TRG TH0)*COS DG(TRG PS0)
      DT21 =COS DG(TRG TH0)*SIN DG(TRG PS0)
      DT31 =-SIN DG(TRG TH0)
      DT12 =SIN DG(TRG PHI)*SIN DG(TRG TH0)*COS DG(TRG PS0)
     1       -COS DG(TRG PHI)*SIN DG(TRG PS0)
      DT22 =SIN DG(TRG PHI)*SIN DG(TRG TH0)*SIN DG(TRG PS0)
     1       +COS DG(TRG PHI)*COS DG(TRG PS0)
      DT32 =SIN DG(TRG PHI)*COS DG(TRG TH0)
      DT13 =COS DG(TRG PHI)*SIN DG(TRG TH0)*COS DG(TRG PS0)
     1       +SIN DG(TRG PHI)*SIN DG(TRG PS0)
      DT23 =COS DG(TRG PHI)*SIN DG(TRG TH0)*SIN DG(TRG PS0)
     1       -SIN DG(TRG PHI)*COS DG(TRG PS0)
      DT33 =COS DG(TRG PHI)*COS DG(TRG TH0)
C
C     COMPUTE TXI0,TYI0,TZI0 (ORIGIN FOR ROTATION)
C
      XP = 0.
      YP = 0.
      ZP = -TRG R
C
      TXI0 = DT11*XP + DT12*YP + DT13*ZP + TXO
      TYI0 = DT21*XP + DT22*YP + DT23*ZP + TYO
      TZI0 = DT31*XP + DT32*YP + DT33*ZP + TZO
```

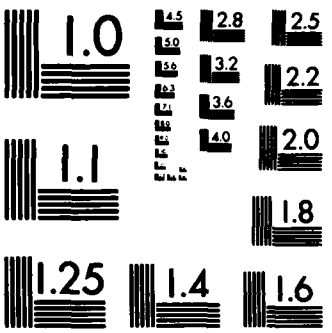
```
C
C       RESET TRG PS0 FOR NEW SYSTEM
C
        TRG PS0 = 0.
C
        YP  = 0.
        VYP = 0.
        AYP = 0.
C
        JMAN = 1
C
  250 CONTINUE
C
        TRG PS = TRG PS0 + PS DOT*(TIME-OLD T)
C
        ZP  =  TRG R*COS DG(TRG PS)
        VZP =  - TRG V0*SIN DG(TRG PS)
        AZP =  -((TRG V0**2)/TRG R)*COS DG(TRG PS)
C
        XP  =  TRG R*SIN DG(TRG PS)
        VXP =  + TRG V0*COS DG(TRG PS)
        AXP =  -((TRG V0**2)/TRG R)*SIN DG(TRG PS)
C
        TRG XI = DT11*XP + DT12*YP + DT13*ZP
        TRG YI = DT21*XP + DT22*YP + DT23*ZP
        TRG ZI = DT31*XP + DT32*YP + DT33*ZP
C
        TRG VXI = DT11*VXP + DT12*VYP + DT13*VZP
        TRG VYI = DT21*VXP + DT22*VYP + DT23*VZP
        TRG VZI = DT31*VXP + DT32*VYP + DT33*VZP
C
        TRG AXI = DT11*AXP + DT12*AYP + DT13*AZP
        TRG AYI = DT21*AXP + DT22*AYP + DT23*AZP
        TRG AZI = DT31*AXP + DT32*AYP + DT33*AZP
C
        TRG XI = TRG XI + TXI0
        TRG YI = TRG YI + TYI0
        TRG ZI = TRG ZI + TZI0
C
C       **********************************************
C       ****                                      ****
C       ****   Relative Target-Missile Pos & Vel  ****
C       ****                                      ****
C       **********************************************
C
  300 CONTINUE
C
        XI HT  =  TRG XI - MSL XI
        YI HT  =  TRG YI - MSL YI
        ZI HT  =  TRG ZI - MSL ZI
```

```
C
      RNG XI =  XI HT
      RNG YI =  YI HT
      RNG ZI =  ZI HT
C
      RNG I  = DSQRT((RNG XI**2)+(RNG YI**2)+(RNG ZI**2))
C
C     CHECK RNG I
C
      IF(DABS(RNG I).LT.MISS DS)HIT=.TRUE.
      IF(DABS(RNG I).LT.MISS DS)WRITE(1,1010)
      IF(DABS(RNG I).LT.MISS DS)RETURN
C
      VXI HT  =  TRG VXI - MSL VXI
      VYI HT  =  TRG VYI - MSL VYI
      VZI HT  =  TRG VZI - MSL VZI
C
      VC  =  (TRG VXI - MSL VXI) * (RNG XI)
     1      +(TRG VYI - MSL VYI) * (RNG YI)
     2      +(TRG VZI - MSL VZI) * (RNG ZI)
      VC   =  VC/RNG I
C
C     CHECK VC
C
      IF(VC .EQ.0.)HIT=.TRUE.
      IF(VC .EQ.0.)WRITE(1,1000)
      IF(VC .EQ.0.)RETURN
C
C     DETERMINE IF HIT
C
      IF(RNG I - DABS(VC)*STEP.LE.MISS DS)HIT=.TRUE.
C
C     CALCULATE THE INTERCEPT ASPECT ANGLE
C
      VXB HT = D11*TRG VXI + D12*TRG VYI + D13*TRG VZI
      VYB HT = D21*TRG VXI + D22*TRG VYI + D23*TRG VZI
      VZB HT = D31*TRG VXI + D32*TRG VYI + D33*TRG VZI
      VBXZ   = DSQRT((VXB HT**2)+(VZB HT**2))
      VB HT  = DSQRT((VBXZ**2)+(VYB HT**2))
C
      INT ASP = DATAN (VZB HT/VXB HT)
      INT ASP = INT ASP + GMB THT
      IF(VXB HT.LT.0.) INT ASP = PI - INT ASP
C
C     ************************************************
C     ****                                        ****
C     ****          Seeker Module                 ****
C     ****                                        ****
C     ************************************************
C
```

```
  400 CONTINUE
C
C     SEEKER ANGLES (BODY TO SEEKER)
C
      SEEK 11  =  DCOS (GMB THT) * DCOS (GMB PSI)
      SEEK 12  =  DCOS (GMB THT) * DSIN (GMB PSI)
      SEEK 13  =- DSIN (GMB THT)
      SEEK 21  =- DSIN (GMB PSI)
      SEEK 22  =  DCOS (GMB PSI)
      SEEK 23  =  0.
      SEEK 31  =  DSIN (GMB THT) * DCOS (GMB PSI)
      SEEK 32  =  DSIN (GMB THT) * DSIN (GMB PSI)
      SEEK 33  =  DCOS (GMB THT)
C
      RNG XB = D11*RNG XI + D12*RNG YI + D13*RNG ZI
      RNG YB = D21*RNG XI + D22*RNG YI + D23*RNG ZI
      RNG ZB = D31*RNG XI + D32*RNG YI + D33*RNG ZI
C
      RNG BXY= DSQRT((RNG XB**2)+(RNG YB**2))
      RNG BZY= DSQRT((RNG ZB**2)+(RNG YB**2))
C
C     COMPUTE CORRECT LOS ANGLES
C
      EP = DATAN2 (- RNG ZB,RNG BXY)
      EY = DATAN2 (RNG YB,RNG XB)
C
      SEEK EP = - GMB THT + EP
      SEEK EY = - GMB PSI + EY
C
      DLOSQ = (20.) * SEEK EP
      DLOSR = (20.) * SEEK EY
C
      GMB WY = SEEK 21*PM + SEEK 22*QM + SEEK 23*RM
      GMB WZ = SEEK 31*PM + SEEK 32*QM + SEEK 33*RM
C
C     ************************************************
C     ****                                        ****
C     ****              Time to Go                ****
C     ****                                        ****
C     ************************************************
C
  500 CONTINUE
C
      IF(TGO HT.EQ.0.)HIT=.TRUE.
      IF(TGO HT.EQ.0.)WRITE(1,1030)TIME,TGO HT
      IF(TGO HT.EQ.0.)RETURN
C
      IF(JTIME.EQ.2) GOTO 510
C
C     RIGG'S TGO
```

AD-A132 491 BANK-TO-TURN CONTROL(U) AIR FORCE INST OF TECH 2/2
WRIGHT-PATTERSON AFB OH D J CAUGHLIN 1983
AFIT/CI/NR-83-25T

UNCLASSIFIED F/G 16/4.1 NL



```
C
      AHAT = (AMAX *(DURAT-TIME)+
     1       AMIN*(TGO HT +TIME-DURAT))/TGO HT
      IF(TIME.GT.DURAT) AHAT = AMIN
      A INC = (VC**2) + 4.0*AHAT*RNG I
      IF(A INC.LT.0.) A INC = VC ** 2
      TGO HT = 2. * RNG I/(DABS(VC) + DSQRT(A INC))
C
      GOTO 600
C
  510 CONTINUE
C
C     MODIFIED RIGG'S TGO
C
      AHAT = (AMAX *(DURAT-TIME)+
     1       AMIN*(TGO HT +TIME-DURAT))/TGO HT
      IF(TGO HT.LT.DURAT) AHAT = (AMAX *(DURAT-TIME))/TGO HT
      IF(TIME.GT.DURAT) AHAT = AMIN
      AOFF = DATAN2(RNG BZY,RNG XB)
      AHAT = AHAT * DCOS(AOFF)
      A INC = (VC**2) + 4.0*AHAT*RNG I
      IF(A INC.LT.0.) A INC = VC ** 2
      TGO HT = 2. * RNG I/(DABS(VC) + DSQRT(A INC))
C
C     ************************************************
C     ****                                        ****
C     ****          Missile Control Law           ****
C     ****                                        ****
C     ************************************************
C
  600 CONTINUE
C
      OLDU3 = U(3)
      INT THT = 0.0
      INT PSI = 0.0
C
      U(1) = AMAX
      IF(TIME.GT.DURAT) U(1) = AMIN
      U(2) = 0.
      U(3) = 0.
      U(4) = 0.
C
      IF(TIME.LT.DLAY SW) GOTO 700
C
      IF(JCNTRL.EQ.2) GOTO 610
C
C     G1
C
      TGO HT3 = TGO HT * TGO HT * TGO HT
      CC11  = 3.*TGO HT/(3.*GAMMA + TGO HT3)
```

```
      CC12 = CC11 * TGO HT
C
C     ACCEL COMMAND IN FT/SEC**2
C
      MSL AXI  =  (CC11*XI HT + CC12*VXI HT)
      MSL AYI  =  (CC11*YI HT + CC12*VYI HT)
      MSL AZI  =  (CC11*ZI HT + CC12*VZI HT)
C
      X COMND  =  D11*MSL AXI + D12*MSL AYI + D13*MSL AZI
      U(2)     =  D21*MSL AXI + D22*MSL AYI + D23*MSL AZI
      U(3)     =  D31*MSL AXI + D32*MSL AYI + D33*MSL AZI
C
      IF(JCNTRL.EQ.3) GOTO 620
C
      GOTO 640
C
  610 CONTINUE
C
      TP3  =  .05*TGO HT*DCOS(GMB PSI)
      IF(TP3.LT..05) TP3 = .05
C
      U(3)     = (PM*DSIN(GMB PSI) - DLOSQ + GMB WY)*
     1            ((E**(TP3) + E**(-TP3))/(E**(TP3) - E**(-TP3)))
      U(3)     = U(3) * MSL VXB / DCOS(GMB PSI)
      U(3)     = DLDU3 + U(3)*.05
C
      GOTO 630
C
  620 CONTINUE
C
      XY CMND = DSQRT((X COMND**2) + (U(2)**2))
      INT THT = DATAN2(-U(3),XY CMND)
      INT THT = GMB THT + (DLOSQ/20.) - INT THT
      INT PSI = DATAN2(U(2),DABS(X COMND))
      INT PSI = GMB PSI + (DLOSR/20.) - INT PSI
      TP3  =  .05*TGO HT
      IF(TP3.LT..1) TP3 = .1
C
      U(3)     = -GMB THT*
     1            ((E**(TP3) + E**(-TP3))/(E**(TP3) - E**(-TP3)))
      U(3)     = U(3) + 2.*INT THT/(E**(TP3) - E**(-TP3))
      U(3)     = U(3) * MSL VXB / DCOS(GMB PSI)
      U MAX    = U(3) * (NY MAX / U(2))
      IF(DABS(U(3)).GT.U MAX) U(3) = DSIGN(U MAX,U(3))
C
      TP2  =  TP3
      U(2)     =  +GMB PSI *((E**(TP2) + E**(-TP2))
     1                      /(E**(TP2) - E**(-TP2)))
      U(2)     = U(2) - 2.*INT PSI/(E**(TP3) - E**(-TP3))
      U(2)     = U(2) * MSL VXB / DCOS(GMB THT)
```

```
C
      TP4  = .2*TGO HT
      IF(TP4.LT..2) TP4 = .2
C
      U(4)      =  + GMB PSI *((E**(TP4) + E**(-TP4))
     1                        /(E**(TP4) - E**(-TP4)))
      U(4)      = U(4) - 2.*INT PSI/(E**(TP4) - E**(-TP4))
C
      GOTO 650
C
  630 CONTINUE
C
      TP2  =  TGO HT
      IF(TP2.LT..2) TP2 = .2
      U(2)      =  +GMB PSI *((E**(TP2) + E**(-TP2))
     1                        /(E**(TP2) - E**(-TP2)))
      U(2)      = U(2) * MSL VXB / DCOS(GMB THT)
C
  640 CONTINUE
C
C     ACCEL COMMAND IN DEG/SEC < P MAX
C
      TP4  = .4*TGO HT
      IF(TP4.LT..2) TP4 = .2
      SN THT = DSIN (GMB THT)
      CN THT = DCOS (GMB THT)
      SN PSI = DSIN (GMB PSI)
      CN PSI = DCOS (GMB PSI)
      K = - (CN THT * SN PSI)**2 - (SN THT**2)
      SCALE = CN PSI * DSQRT(((SN PSI**2) + K)/((SN PSI**2) -1.))
      U MAX = P MAX * SCALE
C
      U(4)      =  + GMB PSI *((E**(TP4) + E**(-TP4))
     1                        /(E**(TP4) - E**(-TP4)))
      U(4)      = U(4) - 2.*INT PSI/(E**(TP4) - E**(-TP4))
      U(4) = U(4) / SCALE
C
  650 IF(JLIMIT.EQ.0) GOTO 700
C
      IF(DABS(U(2)).GT.NY MAX) U(2) = DSIGN(NY MAX,U(2))
      IF(DABS(U(3)).GT.NZ MAX) U(3) = DSIGN(NZ MAX,U(3))
      IF(JCNTRL.EQ.3) U(4) = U(4) - U(3)*
     1DSIN(GMB PSI)/(MSL VXB*DCOS(GMB PSI))
      IF(DABS(U(4)).GT.P MAX) U(4) = DSIGN(P MAX,U(4))
C
C     **************************************************
C     ****                                          ****
C     ****  Equations of Motion for the Missile     ****
C     ****                                          ****
C     **************************************************
```

```
C
  700 CONTINUE
C
      DT  = THETA * RD TO DG
      DP  = PSI * RD TO DG
      DPH = PHI * RD TO DG
      DGT = GMB THT * RD TO DG
      DGP = GMB PSI * RD TO DG
      DGIP= INT PSI * RD TO DG
      DGIT= INT THT * RD TO DG
      DGIA= INT ASP * RD TO DG
      U2  = U(2) / GRAV
      U3  = - U(3) / GRAV
      U4  = U(4) * RD TO DG
      WRITE(1,1030)TIME,TGO HT
       WRITE(1,1040)RNG I,DGIA
       WRITE(1,1050)DGT,DGP
       WRITE(1,1060)DGIT,DGIP
       WRITE(1,1065)U3,U2,U4
       WRITE(1,1070)DPH,DT,DP
       WRITE(1,1080)MSL XI,MSL YI,MSL ZI
C
C     LINEAR INPUT SIGNS ON (4,5,6) OPPOSITE TO THE
C     BANK TO TURN CONTROL LAW FORMULATION
C
      YPRIME(1) = YONE(4)
      YPRIME(2) = YONE(5)
      YPRIME(3) = YONE(6)
      YPRIME(4) = + D11*U(1) + D21*U(2) + D31*U(3)
      YPRIME(5) = + D12*U(1) + D22*U(2) + D32*U(3)
      YPRIME(6) = + D13*U(1) + D23*U(2) + D33*U(3)
      YPRIME(7) = 0.
      YPRIME(8) = YONE(11)
      YPRIME(9) = YONE(12)
      YPRIME(10)= YONE(13)
C
      YPRIME(14)= (DLOSQ - GMB WY)
      YPRIME(15)= (DLOSR - GMB WZ)
C
      IF(DABS(THETA-(PI/2)).LE.DG TO RD) GOTO 710
C
      YPRIME(11)= - W PHI   *  (YONE(11)-A12*U(2)+A11*U(3)-U(4))
      YPRIME(12)= - W THETA * (YONE(12) + A22*U(2) + A21*U(3))
      YPRIME(13)= - W PSI   *  (YONE(13) - A32*U(2) + A31*U(3))
C-    YPRIME(11)= 0.0
C-    YPRIME(12)= 0.0
C-    YPRIME(13)= 0.0

      RETURN
C
```

```
  710 CONTINUE
C
      YPRIME(11)= - W PHI   * (YONE(11) - U(2)/MSL VXB)
      YPRIME(12)= - W THETA * (YONE(12) + (U(3)/MSL VXB))
      YPRIME(13)= - W PSI   * (YONE(13) + U(4))
C-     YPRIME(11)= - W PHI   * ( - U(2)/MSL VXB)
C-     YPRIME(12)= - W THETA * ( + (U(3)/MSL VXB))
C-     YPRIME(13)= - W PSI   * ( + U(4))
C
      RETURN
C
 1000 FORMAT(1H2,'VC = 0')
 1010 FORMAT(1H2,'RNG I = 0')
 1030 FORMAT(1H2,'TIME = ',E12.4,5X,'TGO HT =      ',E12.4)
 1040 FORMAT(1H2,'Range= ',E12.4,5X,'Aspect Angle= ',E12.4)
 1050 FORMAT(1H2,'Gimbal Theta     = ',E12.4,
     12X,'Gimbal Psi       = ',E12.4)
 1060 FORMAT(1H2,'Intercept Theta  = ',E12.4,
     12X,'Intercept Psi    = ',E12.4)
 1065 FORMAT(1H2,'Nz   = ',E12.4,
     14X,'Ny  = ',E12.4,
     24X,'P   = ',E12.4)
 1070 FORMAT(1H2,'Phi  = ',E12.4,
     12X,'Theta = ',E12.4,
     22X,'Psi  =  ',E12.4)
 1080 FORMAT(1H2,'Xm   = ',E12.4,
     12X,'Ym    = ',E12.4,
     22X,'Zm   =  ',E12.4,/)
      END
```

```
      SUBROUTINE INIT(JOB,HIT)
      IMPLICIT REAL*8 (A-I,K-Z)
      LOGICAL HIT
      REAL PLOTX1,PLOTX2,PLOTY1,PLOTY2,PLOTZ1,PLOTZ2
      REAL MINX,MINY,MAXX,MAXY
      INTEGER I,N,JSTATE,JINPUT,JCNTRL,JTIME,JLIMIT,JROLL,JOB
      INTEGER INCHX,INCHY,NPLOT,NPTS
      COMMON /PLOT/PLOTX1(25),PLOTX2(25),PLOTY1(25),PLOTY2(25),
     1PLOTZ1(25),PLOTZ2(25),NPLOT,NPTS
      COMMON /TRGT/TRG XI,TRG YI,TRG ZI,
     1TRG VXI,TRG VYI,TRG VZI,
     2TRG AXI,TRG AYI,TRG AZI
      COMMON /SET/ALTID0,GMB TH0,GMB PS0,RNG0,TRG V0,
     1ASP TH0,ASP PS0,MSL V0,THET0,PHI0,PSI0,TH0,CSPSI0,SNPSI0,
     2SNTH0,MISS DS,STEP,DLAY SW,TRG XI0,TRG YI0,TRG ZI0,
     3TXO,TYO,TZO,OLD T,TRG PS0,TRG TH0,TRG PHI,JMAN
      COMMON /MSL/Y(15),U(4),TGO HT,RNG I,VC
      COMMON /OUT/W1(25,25)
      COMMON /SEEK/ SEEK 11,SEEK 12,SEEK 13,SEEK 21,
     1SEEK 22,SEEK 23,SEEK 31,SEEK 32,SEEK 33,DLOSQ,DLOSR
      COMMON /OPTION/ JCNTRL,JTIME,JLIMIT,JROLL
C
C     TRIG FUNCTIONS FOR DEGREES
C
      SIN DG(X)   = DSIN(DG TO RD * X)
      COS DG(X)   = DCOS(DG TO RD * X)
C
      IF(JOB.EQ.1) GOTO 1
C
C     VSOUND AT 10,000
C
      VSOUND    = 637.43
      K TO FT   = 1.68894
      DG TO RD  = .0174532925
      RD TO DG  = 57.29577951
C
C     Initial Conditions
C
      ALTID0    =  10000.0
      GMB TH0   =  .107
      GMB PS0   = + 40.0
      RNG0      = 3500.0
      TRG V0    = 0.9
      ASP TH0   = 0.0
      ASP PS0   = 180.0
      MISS DS   = 20.
      HIT       = .FALSE.
C
C     Target Manuever
C
```

```
      TRG PHI = -90.
      JMAN = 0
C
C     MISSILE
C
      MSL V0    = 0.9
      THET0     = 0.0
      PHI0      = 0.0
      PSI0      = 0.0
C
C     INERTIAL TO BODY DIRECTION COSINES
C
      D11 = DCOS(THET0)*DCOS(PSI0)
      D12 = DCOS(THET0)*DSIN(PSI0)
      D13 = - DSIN(THET0)
C
      D21 = DSIN(PHI0)*DSIN(THET0)*DCOS(PSI0) -
     1      DCOS(PHI0)*DSIN(PSI0)
      D22 = DSIN(PHI0)*DSIN(THET0)*DSIN(PSI0) +
     1      DCOS(PHI0)*DCOS(PSI0)
      D23 = DSIN(PHI0)*DCOS(THET0)
C
      D31 = DCOS(PHI0)*DSIN(THET0)*DCOS(PSI0) +
     1      DSIN(PHI0)*DSIN(PSI0)
      D32 = DCOS(PHI0)*DSIN(THET0)*DSIN(PSI0) -
     1      DSIN(PHI0)*DCOS(PSI0)
      D33 = DCOS(PHI0)*DCOS(THET0)
C
      Y(1)      = 0.
      Y(2)      = 0.
      Y(3)      = -ALTID0
      Y(4)      = MSL V0 * VSOUND * K TO FT
      Y(5)      = 0.
      Y(6)      = 0.
      Y(7)      = 0.
      Y(8)      = PHI0
      Y(9)      = THET0
      Y(10)     = PSI0
      Y(11)     = 0.
      Y(12)     = 0.
      Y(13)     = 0.
      Y(14)     = GMB TH0*DG TO RD
      Y(15)     = GMB PS0*DG TO RD
C
C     SEEKER ANGLES (BODY TO SEEKER)
C
      SEEK 11   =  COS DG(GMB TH0) * COS DG(GMB PS0)
      SEEK 12   =  COS DG(GMB TH0) * SIN DG(GMB PS0)
      SEEK 13   =- SIN DG(GMB TH0)
      SEEK 21   =- SIN DG(GMB PS0)
```

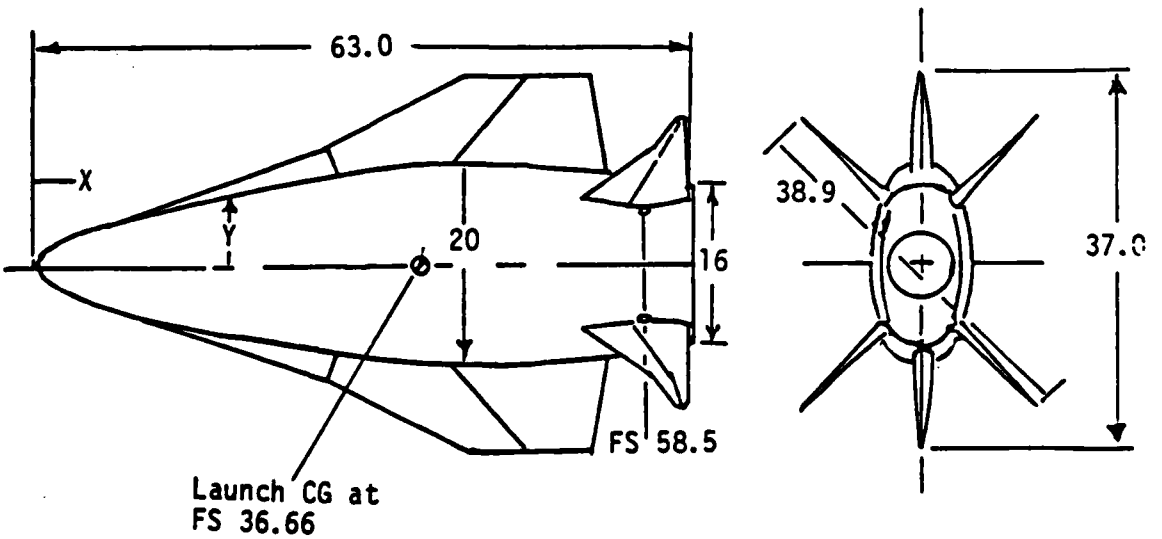
```
      SEEK 22  =  COS DG(GMB PS0)
      SEEK 23  =  0.
      SEEK 31  =  SIN DG(GMB TH0) * COS DG(GMB PS0)
      SEEK 32  =  SIN DG(GMB TH0) * SIN DG(GMB PS0)
      SEEK 33  =  COS DG(GMB TH0)
C
      DLOSQ = 0.
      DLOSR = 0.
C
C     TIME TO GO
C
      TGO HT   =  2.520
C
C     MISSILE CONTROL INPUTS
C
      U(1)     = 0.
      U(2)     = 0.
      U(3)     = 0.
      U(4)     = 0.
C
C     TARGET (BODY TO INERTIAL)
C
      TRG XI   = RNG0*(D11*SEEK11 + D21*SEEK12 + D31*SEEK13)
      TRG YI   = RNG0*(D12*SEEK11 + D22*SEEK12 + D32*SEEK13)
      TRG ZI   = RNG0*(D13*SEEK11 + D23*SEEK12 + D33*SEEK13)
     1           - ALTID0
C
      TRG XI0  = TRG XI
      TRG YI0  = TRG YI
      TRG ZI0  = TRG ZI
C
      TRG AXI = 0.0
      TRG AYI = 0.0
      TRG AZI = 0.0
C
      TRG VXS =  TRG V0*COS DG(ASP TH0) * COS DG(ASP PS0)
      TRG VYS =  TRG V0*COS DG(ASP TH0) * SIN DG(ASP PS0)
      TRG VZS =  TRG V0*SIN DG(ASP TH0)
C
      TRG VXB =  TRG VXS*SEEK11+TRG VYS*SEEK21+TRG VZS*SEEK31
      TRG VYB =  TRG VXS*SEEK12+TRG VYS*SEEK22
      TRG VZB =  TRG VXS*SEEK13+TRG VYS*SEEK23+TRG VZS*SEEK33
C
      TRG VXI =  D11*TRG VXB+D21*TRG VYB+D31*TRG VZB
      TRG VYI =  D12*TRG VXB+D22*TRG VYB+D32*TRG VZB
      TRG VZI =  D13*TRG VXB+D23*TRG VYB+D33*TRG VZB
C
C     CONVERT TO FT/SEC
C
      TRG VXI =  TRG VXI * VSOUND * K TO FT
```

```
      TRG VYI =  TRG VYI * VSOUND * K TO FT
      TRG VZI =  TRG VZI * VSOUND * K TO FT
C
      TRG V0  = TRG V0 * VSOUND * K TO FT
C
C     INITIALIZE RANGE AND VC
C
      RNG I   =  RNG0
      VC      =  (TRG VXI - Y(4)) * DABS(TRG XI)
     1          +TRG VYI * DABS(TRG YI)
     2          +TRG VZI * DABS(TRG ZI)
      VC      =  VC/RNG I
C
C     INITIALIZE TARGET MOTION
C
      IF(TRG VYI.EQ.0) TRG PS0 = 10**3
      IF(TRG VYI.NE.0) TRG PS0 = TRG VXI/TRG VYI
      TRG PS0 = DATAN(TRG PS0)*RD TO DG
      IF(TRG VXI.GE.0.AND.TRG VYI.GE.0.)
     1           TRG PS0 =  TRG PS0
      IF(TRG VXI.GE.0.AND.TRG VYI.LE.0.)
     1           TRG PS0 =   - TRG PS0
      IF(TRG VXI.LE.0.AND.TRG VYI.GE.0.)
     1           TRG PS0 =   + TRG PS0 + 90.
      IF(TRG VXI.LE.0.AND.TRG VYI.LE.0.)
     1           TRG PS0 =   - TRG PS0 - 90.
      TH0     = DSQRT((TRG VXI**2)+(TRG VYI**2))
      IF(TH0.EQ.0) TRG TH0 = 10**3
      IF(TH0.NE.0) TRG TH0 = TRG VZI/TH0
      TRG TH0 = DATAN(TRG TH0)*RD TO DG
      IF(TRG VZI.GE.0.) TRG TH0 =  - TRG TH0
      CSPSI0 = COS DG(TRG PS0)
      SNPSI0 = SIN DG(TRG PS0)
      SNTH0  = SIN DG(TH0)
      TXO    = TRG XI
      TYO    = TRG YI
      TZO    = TRG ZI
      OLD T  = 0.0
C
      RETURN
C
C     OUTPUT
C
    1 CONTINUE
C
      TRG ALT  = -TRG ZI
      TRG PH0  = 0.
C
      PHIm   = Y(8)*RD TO DG
      THETAm = Y(9)*RD TO DG
```

```
      PSIm    = Y(10)*RD TO DG
      WRITE(NPLOT,100)
  100 FORMAT(1H1,////1X,13('*'),' Bank-to-Turn Missile ',
     1'Simulation ',12('*'),//)
      WRITE(NPLOT,200) ALTID0,MSL V0,THETAm,PHIm,
     1GMB TH0,GMB PS0,ASP TH0,ASP PS0,TRG ALT,TRG V0,
     2TRG TH0,TRG PH0,RNG0
  200 FORMAT(1X,'Inital Set Up',///,
     15X,'Missile :'//,15x,'Altitude = ',G12.2,
     25X,'Velocity =  ',G12.2,/,15X,'Pitch    =',G12.2,
     36X,'Roll     = ',G12.2,/,15x,'Gym Theta= ',G12.2,
     45X,'Gym Psi  =',G12.2,//,5X,'Target :'//,
     515x,'Asp Theta=',G12.2,6X,'Asp Psi  = ',G12.2,/,
     615x,'Altitude = ',G12.2,5X,'Velocity = ',G12.2,/,
     715X,'Pitch    = ',G12.2,5X,'Roll     =',G12.2,//,
     815X,'Range    = ',G12.2,/)
      WRITE(NPLOT,300) Y(4),Y(5),Y(6),TRG XI,TRG YI,TRG ZI,
     1TRG VXI,TRG VYI,TRG VZI
  300 FORMAT(1H0,'Inital Position/Velocity',//,
     15X,'Missile :'//,15x,'X Velocity     = ',G12.2,/,
     215X,'Y Velocity     =',G12.2,/,15X,'Z Velocity     =',
     3G12.2,//,
     45X,'Target :'//,15x,'X     = ',G12.2,/,
     515X,'Y     = ',G12.2,/,15X,'Z     = ',G12.2,//,
     615x,'X Velocity     = ',G12.2,/,
     715X,'Y Velocity     = ',G12.2,/,15X,'Z Velocity     =',
     8G12.2)
      WRITE(NPLOT,400) MISS DS,TGO HT,TRG PHI
  400 FORMAT(1H0,'Parameters ',//,
     15X,'Miss Distance  = ',G12.2,
     25X,'TGO HT         = ',G12.2,/,
     35X,'Target Roll    = ',G12.2)
      IF(JCNTRL.EQ.1) WRITE(NPLOT,500)
  500 FORMAT(1H0,'G1 Guidance')
      IF(JCNTRL.EQ.2) WRITE(NPLOT,510)
  510 FORMAT(1H0,'Pro-Nav Guidance')
      IF(JCNTRL.EQ.3) WRITE(NPLOT,520)
  520 FORMAT(1H0,'Bank-to-Turn Guidance')
      IF(JLIMIT.EQ.0) WRITE(NPLOT,530)
  530 FORMAT(1H0,'Unconstrained')
      IF(JLIMIT.EQ.1) WRITE(NPLOT,540)
  540 FORMAT(1H0,'Constrained')
      IF(JTIME.EQ.1) WRITE(NPLOT,550)
  550 FORMAT(1H0,'Riggs Tgo')
      IF(JTIME.EQ.2) WRITE(NPLOT,560)
  560 FORMAT(1H0,'New Tgo')
C
      RETURN
      END
```

APPENDIX D
GENERIC MISSILE CHARACTERISTICS

The generic bank-to-turn missile (Figure D.1) is a short range missile that can produce more than 100 g of normal acceleration and approximately 5 g of lateral acceleration. ROLL rates of up to 500 degrees per second are attainable. Launch weight is 165 pounds with 50 pounds of propellant that produces 4712 pounds of thrust for 2.6 seconds. The shape is 2:1 elliptical cross section with a blunt base and parabolic forebody. The wings are planar with a clipped double delta platform. The tails are all movable crusiform also with a clipped double delta. Additional information is available in [30,31].



Dimensions in Inches

Figure D.1 BANK-TO-TURN MISSILE

LIST OF REFERENCES

[1] J.R. McClendon and P.L. Verges, "Applications of Modern Control and Estimation Theory to the Guidance and Control of Tactical Air-to-Air Missiles," Technical Report RG-81-20, "Research on Future Army Modular Missile," US Army Missile Command, Redstone Arsenal, Alabama, March 1981.

[2] S.A. Murtaugh and H.E. Criel, "Fundamentals of Proportional Navigation," IEEE Spectrum, pp. 75-85, December 1966.

[3] L.A. Stockum, and I.C. Weimer, "Optimal and Suboptimal Guidance for a Short Range Homing Missile," IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-12, No. 3, pp. 355-361, May 1976.

[4] G. Hesprich, "Guidance Law Studies," Section 8, Northrop EMD FINCH Study Report, September 1975.

[5] A.E. Bryson and Y.C. Ho, Applied Optimal Control, Blaisdell Publishing Company, Waltham, Massachusetts, 1969.

[6] N.K. Gupta, J.W. Fuller, and T.L. Riggs, "Modern Control Theory Methods for Advanced Missile Guidance," Technical Report RG-81-20, "Research on Future Army Modular Missile," U.S. Army Missile Command, Redstone Arsenal, Alabama, March 1981.

[7] C.F. Price and R.S. Warren, "Performance Evaluation of Homing Guidance Laws for Tactical Missiles," The Analytic Sciences Corporation, Report Number TR-170-4, January 1973.

[8] P.H. Fiske, "Advanced Digital Guidance and Control Concepts for Air-to-Air Tactical Missiles," AFATL-TR-77-130, Air Force Armament Laboratory, United States Air Force, Eglin Air Force Base, Florida, November 1977.

[9] J.N. Youngblood, "Advanced Linear Guidance Laws for Air-to-Air Missiles," Bureau of Engineering Research, University of Alabama, AFATL-TR-80-12, Air Force Armament Laboratory, United States Air Force, Eglin Air Force Base, Florida, January 1980.

[10] M.E. Warren and T.E. Bullock, "Development and Comparison of Optimal Filter Techniques with Application to Air-to-Air Missiles," Electrical Engineering Department, University of Florida, Prepared for the Air Force Armament Laboratory, Eglin Air Force Base, Florida, March 1980.

[11] T.L. Riggs, Jr. and P.L. Verges, "Advanced Air-to-Air Guidance Using Optimal Control and Estimation," AFATL-TR-81-56, Air Force Armament Laboratory, United States Air Force, Eglin Air Force Base, Florida, June 1981.

[12] T.L. Riggs, Jr., "Linear Optimal Guidance for Short Range Air-to-Air Missiles," Proc. NAECON 1979, Vol. II, May 1979.

[13] T. Kaliath, Linear Systems, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1980.

[14] H. Kwakernaak and R. Sivan, Linear Optimal Control Systems, Wiley-Interscience, New York, 1972.

[15] Y. Bar Shalom and E. Tse, "Dual Effect, Certainty Equivalence, and Separation in Stochastic Control," IEEE Trans. on Automatic Control, Vol. AC-19, No. 5, pp. 494-500, October 1980.

[16] H. Van De Water and J.C. Willems, "The Certainty Equivalence Property on Stochastic Control," IEEE Trans. on Automatic Control, Vol. AC-26, No. 5, pp. 1080-1087, October 1981.

[17] L.C. Kramer and M. Athans, "On the Application of Deterministic Optimization Methods to Stochastic Control Problems," IEEE Trans. on Automatic Control, Vol. AC-19, No. 1, pp. 22-30, February 1974.

[18] D.W. Tufts and D.A. Schnidman, "Optimum Waveform Subject to Both Energy and Scalar Value Constraints," Proc. IEEE, Vol. 52, pp. 1002-1007, September 1964.

[19] M. Vidyasagar, Nonlinear Systems Analysis, Prentice-Hall, Englewood Cliffs, New Jersey, 1978.

[20] N.N. Moiseev and F.L. Chernousko, "Asymptotic Methods in the Theory of Optimal Control," IEEE Trans. on Automatic Control, Vol. AC-26, No. 5, pp. 993-1000, October 1981.

[21] B. Stridhar and N.K. Gupta, "Singular Perturbation Based Guidance and Control Laws for Interceptors Against Maneuvering Targets," Technical Report RG-81-20, "Research on Future Army Modular Missile," U.S. Army Missile Command, Redstone Arsenal, Alabama, March 1981.

[22] G. Blankenship, "Singularly Perturbed Difference Equations in Optimal Control Problems," IEEE Trans. on Automatic Control, Vol. AC-26, No. 4, pp. 911-917, August 1981.

[23] G.K.F. Lee, "Effects of Missile and Target Acceleration Parameters on Time-to-Go Estimation," Colorado State University, Fort Collins, Colorado (unpublished).

[24] H. Goldstein, Classical Mechanics, Addison-Wesley Publishing Company, Reading, Massachusetts, 1965.

[25] D.O. Dommasch, S.S. Sherby and T.F. Connolly, Aeroplane Aerodynamics, Pitman Publishing Corporation, New York, 1967.

[26] USAF Test Pilot School, "Stability and Control Flight Test Theory," AFFTC-TIH-77-1, revised February 1977.

[27] J.D. Kendrick, "Estimation of Aircraft Motion Using Pattern Recognition Orientation Measurements," Ph.D. Dissertation, Air Force Institute of Technology, 1978.

[28] R.W. Hamming, Numerical Methods for Scientists and Engineers, McGraw-Hill, New York, 1962.

[29] J.L. Farrell, Integrated Aircraft Navigation, Academic Press, New York, 1976.

[30] "Bank-to-Turn Steering for Tactical Missiles," AFATL-TR-76-150 (CONF), Air Force Armament Laboratory, United States Air Force, Eglin Air Force Base, Florida, 1976.

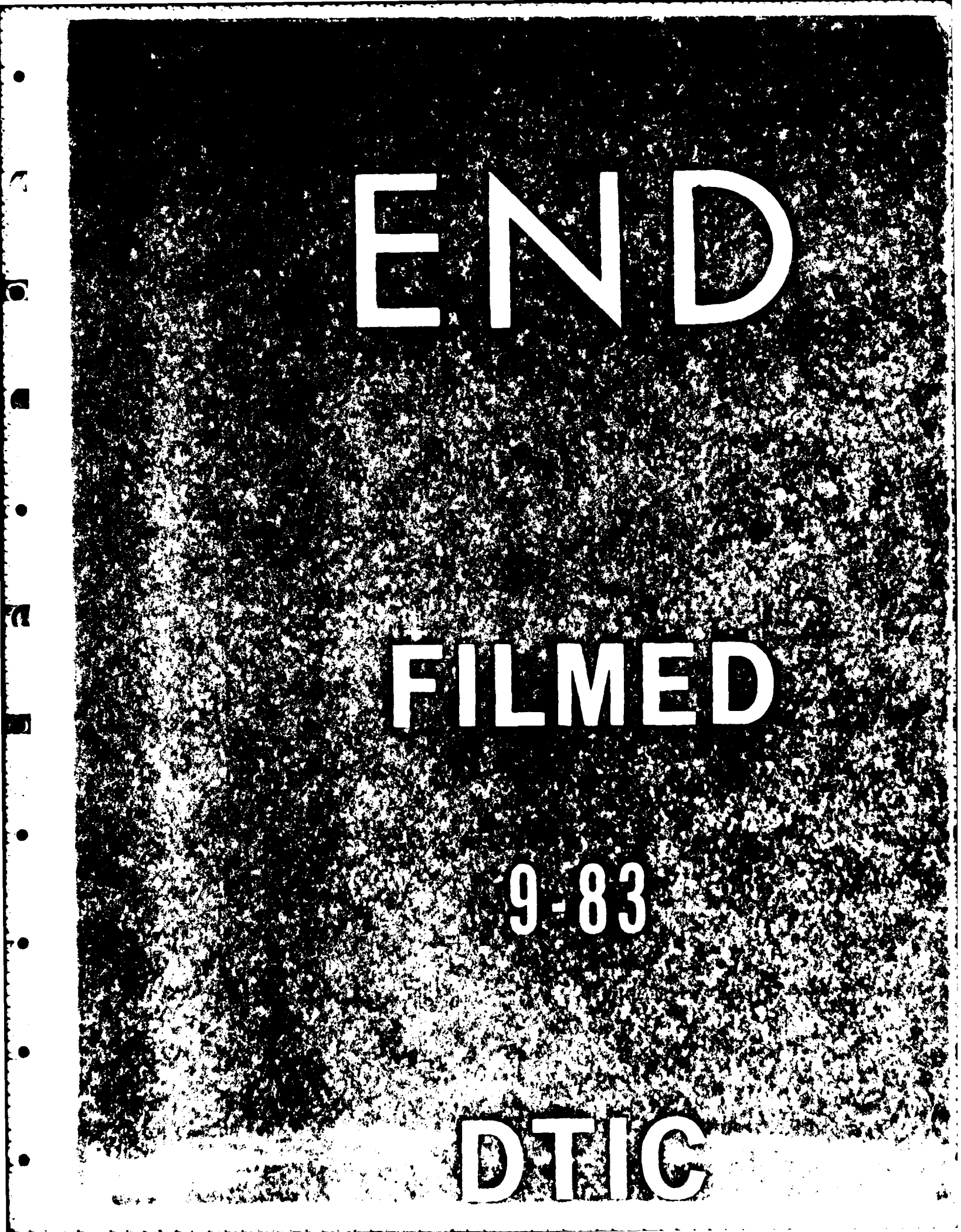
[31] "Bank-to-Turn Configuration Aerodynamic Analysis Report," Rockwell International Report No. C77-1421/034C (CONF), date unknown.

BIOGRAPHICAL SKETCH

Donald J. Caughlin, Jr. was born in San Pedro, CA on Dec. 17, 1946. After graduation from high school, he entered the United States Air Force Academy and was graduated in 1968 as a Second Lieutenant with a B.S. in physics. After graduating with academic honors from Undergraduate Pilot Training, he served as an Instructor Pilot from 1969 to 1971. He attended A-1 upgrade training and flew rescue missions in Southeast Asia as an A-1 "Sandy" pilot until January 1973. Returning to the states, Captain Caughlin flew the F-111F where he was the Wing Weapons Officer and also responsible for avionics software development. Assigned to the Tactical Air Warfare Center, he was the Program Manager and Operational Test Director for the modification and integration of the Pave Tack Infrared system on the F-111F. In December 1978, he completed the USAF Test Pilot School course as a Distinguished Graduate and was reassigned to Eglin AFB, Florida. As a test pilot, Major Caughlin successfully completed numerous aircraft, weapons, and avionics programs in the F-111, F-4, and F-16 aircraft. He was the Armament Division Director for the LANTIRN Night Attack System when he was sponsored by the Air Force Institute of Technology for graduate school. Major Caughlin will be returning to the USAF Test Pilot School as an instructor.

Major Caughlin is a member of the Society of Experimental Test Pilots and a student member of IEEE.

Major Caughlin is married to the former Barbara Schultz of Montgomery, AL. They have two children, a daughter, Amy Marie, age four, and Jon Andrew, age four months.


END
FILMED
9-83
DTIC